Numerical methods in soil-structure interaction.

Reed, M. B.

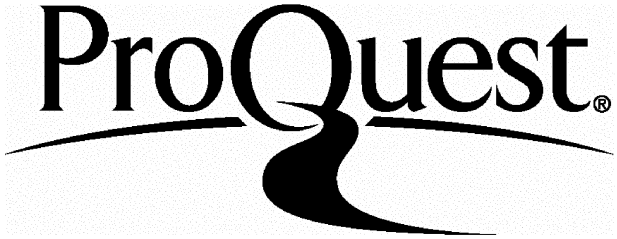
Award date:
1980

Awarding institution:
University of Bath

NUMERICAL METHODS IN SOIL-STRUCTURE INTERACTION

Submitted by M.B.Reed for the degree of Ph.D.
of the University of Bath
1980

Mark Reed



Summary

In this thesis we develop the theory for two separate computer programs capable of modelling the plane strain consolidation of a soil layer under a variety of types of surface loading. We consider loading from flexible footings, rigid footings, built-up embankments and a plane frame structure on individual footings.

The first program uses a simplified form of finite element analysis for the frame structure, and a method of discretizing the surface loading into a series of line-loads. A stress distribution theory applicable to a soil layer on a smooth rigid base is then combined with classical elasticity theory and a finite difference solution of the two-dimensional Terzaghi consolidation equation.

The second program uses a unified finite element analysis modelling both structure and soil, and incoporating the Biot theory of consolidation. A process of smoothing the immediate nodal excess pore-pressures is developed, which allows standard types of finite element to be used in the soil model.

In addition, it is shown how a set of data of void ratio against pressure from a compression test may be analysed (using smoothing splines) with a desk top computer to yield an estimate of the pre-consolidation pressure of a soil sample.

Numerical results from both programs are presented and compared for a number of loading problems, and it is concluded that the finite difference program is considerably more efficient in the solution of problems involving homogeneous soils and loading from flexible and rigid footings; in contrast, the unified finite element analysis has advantages in the solution of more complex problems.

Acknowledgements

I should like to thank the Science Research Council and the University of Bath for their financial support during the course of this project.

I am grateful for the interest and support I have received from many quarters, and particularly to my supervisors Mr D A Cook and Mr F B Ellerby for their guidance, and to Dr D J Naylor for permission to use the finite element package FINEPAK.

My thanks also to Mrs S E Candy and Miss J M Gingell for typing the manuscript.

INDEX

Notation

Notation in this thesis is defined where it first appears; a list of the more commonly employed notation is given here. (Notation specific to finite element theory is marked f.e.)

A, B	Skempton pore pressure coefficients
A_p	plane strain pore pressure coefficient
a	footing half-width
$\underset{\sim}{a}$	vector of forces and pore-pressure reactions (f.e.)
B	strain-displacement matrix (f.e.)
$\underset{\sim}{b}$	vector of body forces (f.e.)
C_c	compression index
C_u	undrained strength
C_v	coefficient of consolidation
c	cohesion (hyperbolic model)
c	scaling factor for pore pressures (f.e.)
D	stress-strain matrix (f.e.)
D	depth of soil layer
E	Young's modulus
E_t	initial tangent modulus (hyperbolic model)
E_u	undrained modulus
F, f	applied forces
G	shear modulus
g	smoothness parameter in smoothing spline
h	layer thickness
I	Moment of Inertia
K	stiffness matrix (f.e.)
K	bulk modulus
K_f	pore-water bulk modulus (f.e.)
K_o	coefficient of lateral earth pressure.

k	permeability
l	length
M	moment
m	parameter controlling rigidity of footing
M_v	compressibility
N_i, N	shape functions and matrix (f.e.)
n	porosity (f.e.)
P	applied load
p	applied pressure
p	pore pressure (f.e.)
p_c	preconsolidation pressure
$\underset{\sim}{q}$	force vector (f.e.)
R_f	correcting constant (hyperbolic law)
s, t	element local coordinate system (f.e.)
s	soil compressive strength (hyperbolic model)
t	time
u	horizontal displacement (f.e.)
u	pore pressure
U, V	mesh-point pore pressure (finite differences)
v	out-of-plane displacement
$\underset{\sim}{v}$	vector of displacements and pore pressures (f.e.)
w	vertical displacement
x	horizontal direction
y	out-of-plane direction
z	vertical direction
α	3D Skempton pore pressure coefficient
δ	displacement vector (f.e.)
Δ	increment

ε	direct strain
ϕ	angle of friction
γ	shear strain
γ'	submerged density
γ_w	water density
ν	Poisson's ratio
π	potential energy (f.e.)
σ	direct stress
σ_1, σ_3	major and minor principal stresses
τ	shear stress
θ	angle of rotation
θ	time discretization (f.e.)

In general, a dashed superscript ' indicates the effective stress value (although this may be omitted where the meaning is clear).

Throughout this thesis, the 'compression positive' sign convention is used.

CHAPTER 1 - INTRODUCTION

1.1 Background: Models of Soil Behaviour

Before any physical problem can be solved numerically, a mathematical analogue of the problem must be produced. This mathematical model will incoporate theories describing the relevant behaviour of the constituent materials of the problem. There are many factors hindering the development of a comprehensive scientific theory of soil behaviour, which we shall first summarize.

Experimental data can be drawn from two sources: full-scale field tests and site records from construction projects, and small-scale laboratory tests and models.

In site investigations, the complex and ill-defined geometry and composition of soil is first approximated on the basis of borehole logs by a series of homogeneous zones of soil with straight-line boundaries (usually horizontal). The properties of each of these soil types must then be assessed. This is done by taking soil samples for testing in the laboratory. Two questions arise: will the sample have the same properties as the larger soil mass, and are those properties affected by the sampling procedure? The answer to both questions is generally unfavourable. Rocks, stones sand, organic matter and air pockets in the soil mass can make its overall behaviour differ from that of a small sample. This is particularly true of the permeability, which is greatly increased by the presence of small layers of sand or holes caused by rootlets. One typical study (Green and Hight (1975)) found laboratory measurements of the coefficient of consolidation C_v to be one to two orders of magnitude smaller than those of the soil in the field. Disturbance of the soil during sampling, on the other hand, will affect its elastic behaviour, affecting the measured values of undrained strength C_u and compressibility M_v.

These errors may be reduced to some extent by taking large diameter piston samples; an alternative procedure which is proving successful is the measurement of soil properties in situ, by field vane tests of strength, field piezometer tests of permeability and pressuremeter tests.

Assuming accurate measurements of the soil parameters, the next step is their incorporation into a theory of soil behaviour, that is a constitutive law linking stresses, strains and volume change. The simplest law, that of isotropic linear elasticity, is a starting point, its main deficiency is its inability to model the concept of failure. Failure of the soil, (where large irrecoverable strains occur once the stresses reach a certain limit) is clearly of major importance in practical applications, and modern theoretical soil mechanics from the original textbook of Terzaghi (1943) to the mid-nineteen fifties was primarily concerned with bearing capacity and failure criteria. However, the confirmation of analytical theories by field testing was frequently confused by the significant dependence of soil properties on the stress history of the material.

A change in emphasis was initiated at that time by the Cambridge research group under Roscoe. Instead of dealing with natural soil deposits of complex stress history, they worked with remoulded clays and well-graded sands which could be subjected to any desired stress or strain path in specially-designed apparatus. The soil parameters were carefully determined in true triaxial conditions, and large model tests carried out using advanced techniques and instrumentation (centrifuge, x-ray photography, electron microscope). As a result of their efforts they developed the theory of critical state. (Roscoe and Burland (1968), Schofield and Wroth (1968) and Roscoe (1970)). In the context of normally and lightly over-consolidated clays in plane strain, such as we are concerned with in this thesis, a simplified theory is given by Burland (1972) in a paper to the 1971 Roscoe Memorial Symposium in Cambridge. Burland's model, of a strain-hardening plastic material with a yield locus (rather than a simple yield point) is valuable, in that it requires knowlege of only four traditional soil parameters: the undrained strength C_u, angle of effective friction $\emptyset'$, compression index C_c and over-consolidation ratio n. Burland conducted model footing tests using a heavily greased steel-framed glass-sided box, and obtained centreline displacements and pore pressures in good agreement with theoretical predictions. In other papers and subsequent discussion in session 5 of the symposium, Gibson and Sills (1972)

produced theoretical results for the deformation of a Gibson soil (an isotropic linear elastic soil whose modulus increases linearly with depth - see Gibson (1974)), results very close qualitatively to those obtained by Zienkiewicz and Naylor (1972) from a finite element model using critical state theory. Similarly, Smith (1970) shows how a simple type of inhomogeneity in a linear elastic isotropic finite element model brought the results into close agreement with observed behaviour of sand and chalk. There is thus a new validation for simpler soil models.

In this thesis the emphasis will be on techniques for the solution of a problem of soil-structure interaction; we shall use elastic soil models with allowance for inhomogeneity and simple non-linearity; this and alternative soil models are discussed further in Chapter 5.

1.2 Soil-Structure Interaction

The Problem

Buildings are normally constructed on one of three types of foundation, sketched in Fig.1.2.1;

(a) individual strip or pad footings

(b) raft foundations of finite stiffness

(c) piled foundations

These three types are listed in increasing order of cost and bearing capacity. As the foundation is loaded during construction, so settlement will occur due to the compression of the underlying soil. In the case of saturated clay soils a large proportion of this settlement will occur after construction has finished, as the pore-water in the soil flows away from the loaded area: the time-dependent process of consolidation.

Clearly, the way in which individual parts of the foundation settle is dependent upon both the nature of the soil below and that of the structure above. The soil and the structure are linked through the foundation, and interact through it. In the case of a framed structure on pad footings, a settlement of one footing more than its neighbours will induce new stresses into the structure, such that load will be removed from that footing and transferred to its neighbours. This load redistribution will in turn affect the individual footing settlement rates.

Until recently such problems of soil structure interaction have been too complex to analyse in routine construction work. The structural engineer would analyse a structure on the assumption that it stood on a perfectly rigid foundation; this analysis would produce the forces and moments applied to the foundation; and the civil engineer would predict foundation settlement on this basis, ignoring the effect of the structure. Settlement predictions obtained in this way will in general overestimate the amount of differential settlement occurring between parts of the foundation. Differential, rather than absolute, settlement is the primary cause of structural damage (see Burland and Wroth (1975)) and tables

of maximum allowable differential settlements have been prepared for practising engineers (Skempton and McDonald (1956)). This uncoupled approach may lead to the final structure being either unnecessarily strong (if it is built to tolerate the predicted settlements) or too weak (if the predicted settlements are not used in reanalysis of the design).

Modern styles of building tend to emphasize the importance of this interaction, and interest in the problem has grown rapidly since Meyerhof (1947) first proposed a technique for its approximate solution. With the advance of computer design and capability, the number of factors which may be taken into account in a mathematical model is always increasing. On the other hand, a recent state of the art report by the Institution of Structural Engineers (1978) recommended (page 30) that "future analytical studies should focus on the need to provide relatively simple aids for design rather than the provision of novel but complex mathematical routines".

In the remainder of this section we review the work that has been done on the problem of soil-structure interaction, with particular reference to the individual footing type of foundation, and in § 1.3 we outline the approach of the present thesis.

Review of Previous Work

Historical reviews of this subject may also be found in Lee (1975) and Wood (1972), chapter 8, among others.

Meyerhof (1947) used slope-deflection equations together with a Winkler soil model (where the soil is treated as a bed of vertical independent linear or non-linear springs, the stiffness being determined from field tests). In this way he analysed one storey two-dimensional frames on individual footings. In a later paper (1953) he showed how a more complicated frame structure could be represented by a beam of equivalent stiffness (his method is outlined in § 3.4). Other analyses based on a Winkler soil model were made by Sved and Kwok (1963), Morris (1966), Lee and Harrison (1970) and Haddadin (1971), the latter using a finite element analysis.

The Winkler soil model is however only a crude approximation, and in Europe research concentrated on the use of a linear elastic soil model, dividing the soil continuum into layers and evaluating surface settlement w by the summation:

$$w = \sum m_v \Delta\sigma_z h \qquad (1.2.1)$$

over all layers, where:

m_v = compressibility of the layer

$\Delta\sigma_z$ = change in vertical stress in the layer, found by Boussinesq stress formulae (see §2.2)

h = layer thickness

Chamecki (1956) produced a way of estimating final settlements of framed structures on footings, as follows:

(i) By structural analysis, calculate the footing loads assuming no settlement.

(ii) Calculate the footing settlements using equation 1.2.1, ignoring the influence of the structure.

(iii) Using load transfer coefficients, find the new footing loads caused by these settlements.

(iv) Reevaluate the settlements on the basis of these new loads.

Chamecki then took the averages of the settlements found in (ii) and (iv) as an approximation to the true behaviour. This became unnecessary when his method was developed into an iterative procedure implemented by computer, by Larnach (1970).

Hain and Lee (1974) made a comparison of the linear elastic and Winkler approaches applied to a raft problem, concluding that the linear elastic model gave more realistic predictions of soil and structural behaviour.

The finite element method first appeared in the mid-1950's as a technique of structural analysis, but it was not until about ten years later that its potential relevance to soil mechanics began to be investigated. In 1965 a study by Cheung and Zienkiewicz of a raft foundation treated the soil as a single continuous element.

Purely soil mechanics applications started with Brown and King (1966) who used the finite element method for the analysis of built-up and cut-down embankments. A good review of the subsequent history of finite elements in soil mechanics is given by Valliappan (1975).

Larnach and Wood have developed two computer programs, capable of analysing framed and raft structures respectively in two and three dimensions, which combine a finite element model of the structure with a non-linear elastic soil model. They improved the evaluation of strains and displacements in the soil by replacing equation 1.2.1 (which assumes no lateral strain) by the formula from three-dimensional elasticity theory:

$$w = \sum \frac{1}{E_t} \left(\Delta\sigma_z - \nu(\Delta\sigma_x + \Delta\sigma_y)\right)h \qquad (1.2.2)$$

where E_t is the Young's modulus of the soil at its current state of stress, as given by the hyperbolic stress-strain model of Duncan and Chang (1970) to be described in § 5.3. Their work is reported in Larnach and Wood (1972) and Wood and Larnach (1975), and full details are to be found in Wood (1972).

A major advance in their programs was the inclusion of time-dependency; that is, they modelled the consolidation process. To do this they used the one-dimensional diffusion equation of Terzaghi:

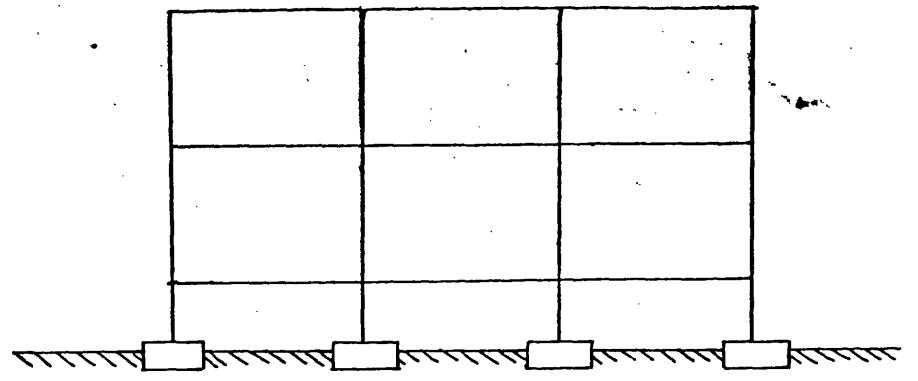
$$\frac{\partial u}{\partial t} = c_v \frac{\partial^2 u}{\partial z^2} \qquad (1.2.3)$$

where u represents excess pore-pressure, and discretized it by an explicit finite difference scheme.

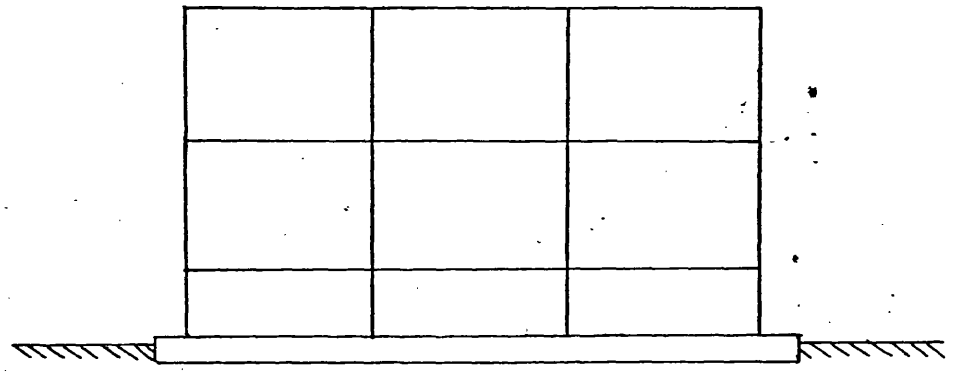
Perhaps the most elegant approach to the problem is the creation of a unified finite element model embracing soil, structural and foundation elements. The advantage of a finite element model for the soil, is that complex soil inhomogeneities, geometries and boundary conditions may be dealt with routinely. Two-dimensional finite element analyses of pile foundations - eg. Hooper (1973), Naylor and Hooper (1975) - and of raft foundations - eg Smith (1970) - have been made recently. Majid and Craig

(1971) analysed a simple frame structure on sand with asymmetric point loads using a three-dimensional finite element model; the full versatility of the finite element method has been demonstrated by King and Chandrasekaran (1975), in a 3-D finite element analysis of a multi-storey multi-bay framed rafted structure on a heterogeneous soil; the same authors (1977) have repeated the analysis using the linear elastic soil model described above.

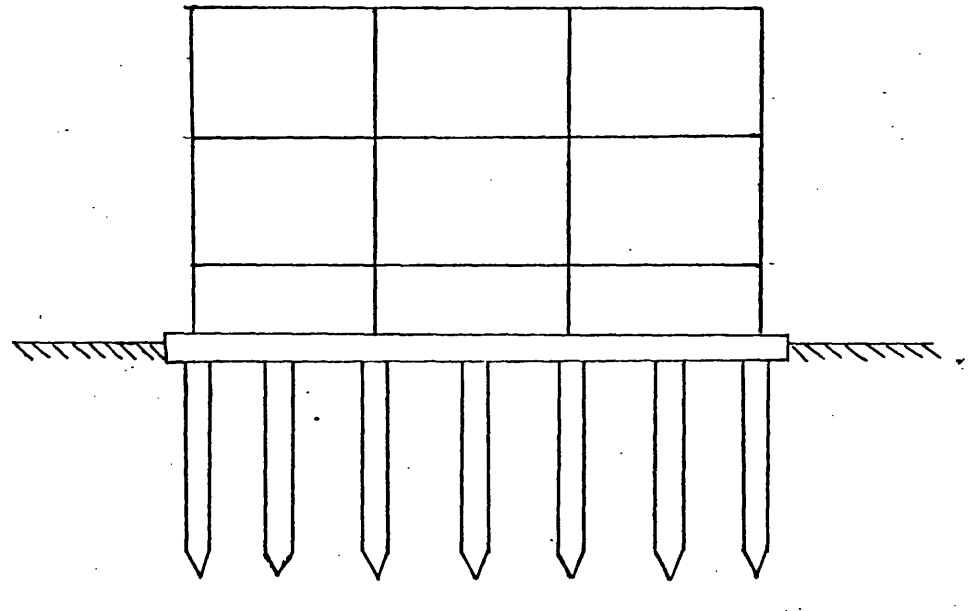
Fuller summaries with results of the more recent works cited will be found in the state of the art report by the ISE (1978).



(a) strip/pad footings



(b) raft foundation



(c) piled foundation

Fig. 1.2.1 Types of foundation

1.3 The Approach of this Thesis

Limitations of current progress

Researchers in the field of soil-structure interaction are agreed on the urgent need for well-documented case histories against which the latest methods of analysis described in the previous section can be tested. Reports of such case studies are starting to appear (see appendix D of the report of the ISE (1978)), although there do not, as yet, appear to have been any model tests of soil-structure interaction problems on simple soils on the lines of the Cambridge research. In the meantime the most useful line of activity would be in improving and expanding the existing methods of solution rather than in the development of new ones. As we have seen in §1.2, the currently-used methods are of two types:

(i) the composite method: finite element model of the structure linked to a layered elastic soil model. This has been expanded to include one-dimensional consolidation by Wood (1972).

(ii) the unified method: finite element model of soil and structure.

We consider the limitations of each in turn:

The composite method (i) has a serious drawback in its use of Boussinesq stress distribution theory in the soil model. This theory, applicable to an ideal elastic half-space, gives considerable errors for the stresses in a finite soil layer with a rigid base. For example, a recent case study found that computed settlements using Boussinesq theory were on average 30% less than those predicted on the basis of stresses from a finite element analysis (Crowser, Schuster and Sack (1975)). Moreover, while the Boussinesq theory gives particularly simple formulae (equations 2.2.9-11) for the stress distribution from a surface line-load, the integration of these formulae to cover other forms of plane strain loading over finite areas increases the complexity considerably. Poulos and Davis (1974) collect the formulae which have been produced for various simple types of loading; other problems, such as the stress distribution from a rigid footing, are not amenable to analytic solution.

The one-dimensional Terzaghi consolidation theory used by Wood has been shown to be a good approximation to the true two or three-dimensional flow situation when the soil layer has a porous base and surface, in which case the bulk of the flow occurs vertically. It is not justified, however, to conclude from this that the one-dimensional theory can be reasonably applied to problems with an impervious base to the soil layer. The horizontal flow would be particularly important in the soil-structure interaction situation, where it acts as a link between the actions of the different parts of the foundation. It is relevant to mention in this context that natural soil deposits have a much higher horizontal than vertical permeability.

The unified method (ii) has great power and versatility of application. It does, however, suffer the inherent drawbacks of the finite element method, namely the need for substantial amounts of computer store and time, the higher mathematical sophistication of its theory, and the extra care and time needed in preparation of input data.

A theory of consolidation by finite elements (see § 4.3) was produced by Sandhu and Wilson (1969), but to the writer's knowledge it has so far only been applied in simple two-dimensional elastic soil problems with no foundation or structural elements included in the model, only special 'composite' soil elements of a particular type.

Aims

In this thesis, the writer considers both the composite and the unified methods of soil-structure interaction analysis, attempting to tackle the limitations of the former while expanding the capabilities of the latter so that direct comparisons of the performance of the two methods can be made. To this end, the goal was set of producing two computer programs (named FDTIM and FINETIM for the composite and unified methods respectively), each capable of performing plane strain consolidation (ie. time-dependent) analyses for shallow and deep soil layers under the following types of surface loading:

(a) flexible footings

(b) built-up embankments

(c) rigid footings

(d) a single-storey multi-bay frame with asymmetric loading.

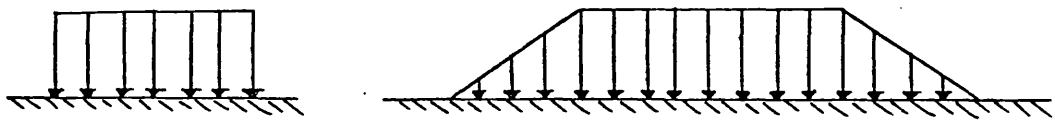
The four types of loading are illustrated in Fig.1.3.1. They were suggested by the availability of analytic and case study solutions against which results could be checked, and by the possibilities of small scale experiments being constructed, instrumented and performed in the soil mechanics laboratory of the School of Architecture and Building Engineering.

A final general aim in the writing of this thesis has been to present a clear, readable and coordinated guide through the relevant fields, steering a path between the misty highlands of mathematical theory and the marshy plains of engineering empiricism.

Summary

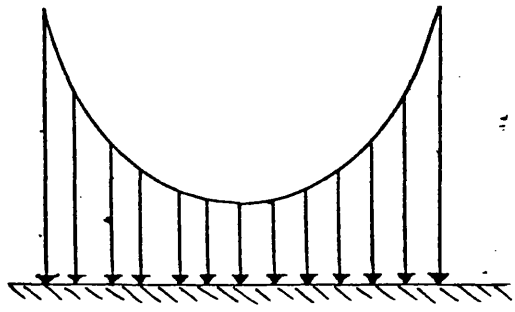
The theory of composite method program FDTIM is developed in Chapters 2 and 3. Chapter 2 presents alternatives to the standard Boussinesq theory for modelling shallow soil layers and finite-area loads, while Chapter 3 deals with two-dimensional consolidation and the structural model. The way in which these theories are combined and implemented in FDTIM is also shown.

Chapter 4 discusses the finite element program FINETIM. Starting from a very general program of Naylor (1977), two-dimensional consolidation theory is incorporated, with an adaptation which allows standard types of soil element to be used. The various models of non-linear soil behaviour are discussed in Chapter 5, in particular the hyperbolic model of Duncan and Chang as used by Wood, Majid and Craig and others. Chapters 6 and 7 give numerical results from both programs on each of the four types of loading problem listed above, and attempt a comparison between their performance. Conclusions are drawn in Chapter 8.

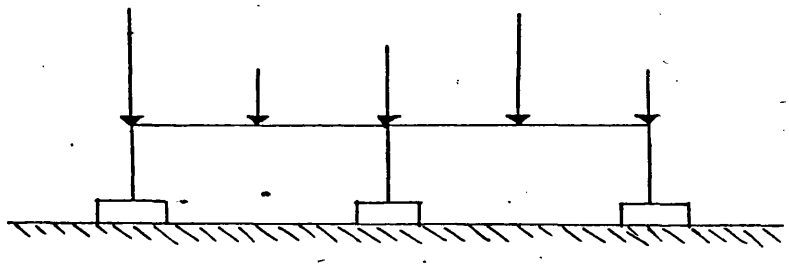


(a) flexible footing

(b) embankment



(c) rigid footing



(d) loaded frame

Fig. 1.3.1 Types of loading

CHAPTER 2: Elasticity Theory and Related Topics

2.1 Introduction

In this chapter we develop the elasticity theory associated with drained and undrained settlement analyses for a linear elastic soil in plane strain with a surface strip load. In § 2.2 the stress increment distributions in the soil for this problem are considered, and the classical Boussinesq theory is adapted to take account of rigidity of the loaded footing, and finite depth of the soil layer.

We then show the way in which this theory may be employed in a numerical model, by representing the soil as a finite rectangular mesh of points. Finally, we consider the excess pore-pressure distribution induced by the applied load, and the relevance of the Skempton pore-pressure coefficients in the plane strain situation.

2.2 The distribution of Stresses in the Soil

The soil at rest

For a soil at rest, ie. with no applied loads acting upon it, the effective vertical stress σ'_{z_o} at the point (x, y, z) is due solely to the weight of soil lying above that point. For a saturated homogenous soil of submerged density γ',

$$\sigma'_{z_o} = \gamma' z \qquad (2.2.1)$$

The horizontal stress at the given point is taken to be a proportion of the vertical stress:

$$\sigma'_{x_o} = \sigma'_{y_o} = K_o \sigma'_{z_o} \qquad (2.2.2)$$

K_o is termed the coefficient of Lateral Earth Pressure, or the coefficient of Earth Pressure at Rest, and lies typically in the range 0.55±0.10 for normally consolidated soils (Poulos (1975)), while for overconsolidated soils K_o increases with increasing overconsolidation ratio. For an ideal elastic soil it may be deduced from the elasticity equations that for no lateral strain:

$$K_o = \frac{\nu}{1 - \nu} \qquad (2.2.3)$$

However in practice K_o is highly dependent on the stress history of the soil, and much work has been done on techniques for its determination (see Harr (1977), p. 269-70); the most widely used empirical relation is:

$$K_o = 1 - \sin \phi' \qquad (2.2.4)$$

first proposed by Jaky (1944), where ϕ' is the effective angle of friction for the soil.

Total stresses may be obtained by adding the hydrostatic pore-water pressure to the effective stresses.

The Boussinesq Problem

The simplest meaningful loading situation one can consider is that of a point load P acting on the surface of a semi-infinte elastic soil mass (fig.2.2.1). This problem was solved by Boussinesq (1885) using a stress function which satisfied the

compatibility and boundary equations. He derived the following formulae (given in rectangular coordinates following Jumikis (1969)):

$$\Delta\sigma_z = \frac{3P}{2\pi}\frac{z^3}{R^5} \qquad (2.2.5)$$

$$\Delta\sigma_x = \frac{3P}{2\pi}\left\{\frac{x^2z}{R^5} - \frac{1-2\nu}{3}\left(-\frac{1}{R(R+z)} + \frac{(2R+z)x^2}{(R+z)^2R^3} + \frac{z}{R^3}\right)\right\} \qquad (2.2.6)$$

$$\Delta\tau_{xz} = \frac{3P}{2\pi}\frac{xz^2}{R^5} \qquad (2.2.7)$$

$$\Delta\tau_{xy} = \frac{3P}{2\pi}\left\{\frac{xyz}{R^5} - \frac{1-2\nu}{3}\left(\frac{(2R+z)xy}{(R+z)^2R^3}\right)\right\} \qquad (2.2.8)$$

$\Delta\sigma_y$ and $\Delta\tau_{yz}$ by symmetry.

These formulae have long been the basis for stress and settlement calculations. They may be integrated to give the stress distributions for line, strip, circular and rectangular loads. Two of these integrated problems are relevant to this study; firstly, an infinite line load of magnitude p/unit length acting along the y-axis, reducing the problem to a two-dimensional plane strain one (fig.2.2.2). The stress components now simplify considerably, notably in becoming independent of Poisson's ratio (Poulos and Davis (1974)):

$$\Delta\sigma_z = \frac{2p}{\pi}\frac{z^3}{r^4} \qquad (2.2.9)$$

$$\Delta\sigma_x = \frac{2p}{\pi}\frac{x^2z}{r^4} \qquad (2.2.10)$$

$$\Delta\tau_{xz} = \frac{2p}{\pi}\frac{xz^2}{r^4} \qquad (2.2.11)$$

These may again be integrated between limits in the x-direction, to yield a solution for the stresses under the edge of an infinite strip load, width a, magnitude p/unit area (fig.2.2.3). The stresses at the point (a/2, z) are (Wood (1972)):

$$\Delta\sigma_z = \frac{p}{\pi}\left(\tan^{-1}a/z + \frac{az}{a^2 + z^2}\right) \qquad (2.2.12)$$

$$\Delta\sigma_x = \frac{p}{\pi}\left(\tan^{-1}a/z - \frac{az}{a^2 + z^2}\right) \qquad (2.2.13)$$

$$\Delta\tau_{xz} = \frac{p}{\pi}\left(1 - \frac{z^2}{a^2 + z^2}\right) \qquad (2.2.14)$$

Using these formulae and the principle of superposition the stresses at any point in the soil can be found.

Analytic solutions for the stress distributions resulting from linear variably-loaded areas, and from horizontal loadings, have been derived, all of this work being collected and summarized by Poulos and Davis (1974).

Where the analytic formula contains, for example, Bessel functions or requires numerical integration, influence tables have been produced for selected values of the parameters. These tables are ideal for hand-calculations, but are unsuitable for use in a computer program. Indeed,even the formulae 2.2.12 and 2.2.13, which can be implemented on the computer using standard functions, proved to be extremely expensive in computer time, due to the inverse tangent function which must be evaluated twice at each mesh-point.

A more efficient method of finding the stresses at any mesh-point in the strip-load problem of fig.2.2.3, is to discretize the strip-load into a series of line-loads, and to sum the influence of each line-load on the mesh-point using equations 2.2.9 - 2.2.11 repeatedly. If the strip-load p/unit area, width a, is divided into N equal substrips, and the load on each substrip lumped into an equivalent line-load in the centre of the substrip, we will have N line-loads of magnitude $\frac{pa}{N}$/unit length, spaced a distance a/N apart. N must be large enough to make a/N small, compared with the distance between mesh-points.

There is one further advantage in this discretization of the load distribution, in that it provides much greater flexibility in attempting to model more complex situations. The two main complexities which arise in this study (and in practical

applications) are:

a. the loaded area being a relatively rigid footing, and

b. the soil consisting of a finite layer underlain by a rigid base, eg. bedrock.

Rigid footing on finite soil layer: contact pressure distribution

Both the above factors will affect the stress distribution in the soil, primarily by influencing the distribution of contact pressure under the footing.

The contact pressure distribution under a relatively rigid footing resting on an elastic infinite half-space has been analysed by Borowicka (1938, 1939), and some of his results are presented in Terzaghi (1943). The pressure distribution curves involve factors such as:

$$K_r = \frac{1}{6}\,\frac{1-\nu^2_{soil}}{1-\nu^2_{footing}}\,\frac{E_{footing}}{E_{soil}}\,\frac{H}{R} \qquad (2.2.15)$$

representing the relative rigidity of a circular footing radius R, thickness H with respect to the soil; for the limiting case of a perfectly rigid strip footing, however, of width 2a with a uniform applied pressure P/unit area as in fig.2.2.4, the contact pressure distribution is:

$$p(x) = \frac{2P}{\pi}\,\frac{1}{\sqrt{1-\left(\frac{x}{a}\right)^2}} \qquad -a<x<a \quad (2.2.16)$$

It should be noted that:

$$\int_{-a}^{a} p(x)\,dx = 2Pa \qquad (2.2.17)$$

ie. the total load on the footing remains constant, but pressure has been redistributed away from the centre towards the edges of the footing. Such a pressure distribution is easy to model by the method of discretization into line-loads described in the previous paragraph, by setting the magnitude of the line-load at x_i equal $p(x_i)\delta_i$ - see fig.2.2.5. However, where the footing overlies a shallow compressible soil layer, equation 2.2.16 exaggerates the redistribution of pressure, and its use in a

settlement analysis produces a hogging of the footing, with the footing edges settling more than the centre.

Indeed, it is to be expected that the shallower the soil depth with respect to the footing,width, the less effect the footing rigidity will have, and the more closely the pressure distribution will approach that for a flexible footing, viz.

$$p(x) = P \qquad -a<x<a \quad (2.2.18)$$

Brown (1969 a,b) has made numerical studies of the contact pressure distribution under a circular raft of radius a and specified rigidity resting on an elastic isotropic soil layer of depth D with a rough rigid base. In his analysis he used the finite-layer stress distribution of Burmister (1956), discretizing the raft into a number of uniform-pressure annuli. Graphs of sample results for shallow, $D/a = 1$, and deep soil layers are given in fig.2.2.7. The contact pressure curves are for $K = 0.1,1,10$ where K is the relative rigidity of the footing.

For the most rigid footing, K = 10, it is seen that as the relative depth of the soil layer is decreased, the contact stress curve becomes increasingly flattened away from the edges, changing from a shape analagous to the Borowicka curve, fig.2.2.4 and approaching the uniform-pressure distribution of a flexible footing.

To model efficiently this phenomenon, we require a family of curves, distinguished by some continuously varying parameter m, of which equations 2.2.16 and 2.2.18 are limiting cases, and for all of which equation 2.2.17 holds.Such a family is given by:

$$p_m(x) = P\, f(m) \left(1 - \left|\frac{x}{a}\right|^m\right)^{-1/m} \qquad -a<x<a \quad (2.2.19)$$

where f(m) is such as to make equation 2.2.17 hold. Thus with $f(2) = \frac{2}{\pi}$ and $f(m)\to 1$ as $m\to\infty$, equation 2.2.19 simplifies to 2.2.16 when m = 2 and approaches 2.2.18 as $m\to\infty$. Since f(m) is independent of x, it does not affect the value of m which gives the minimum differential settlement across a footing for a given relative depth D/a, which we shall denote $m_{opt}\ (D/a)$. The

settlements at $x = 0$ and $x = \frac{1}{2}$ on a rigid loaded strip-footing of half-width $a = 1$ for various depths and values of m, were obtained for $\nu = 0.5$ in an interactive computer program on the MULTICS system based on the elasticity theory described in the following paragraph. From this, fig.2.2.6 of m_{opt} (D/a) against D/a was plotted for the range of relative depths in which we are likely to be interested, viz., $\frac{1}{4} < D/a < 5$, using the finite layer stress distributions developed below. For relative depths greater than 5, the contact pressure distribution is not significantly different from that for a half-space, and for relative depths less than $\frac{1}{4}$ the method becomes inapplicable through the importance of edge-effects, etc. It can be seen that in the range of interest, $2 < m_{opt} < 6$.

We now return to the question of finding f(m). From 2.2.17 and 2.2.19,

$$f(m) = \frac{1}{I_m} \qquad (2.2.20)$$

Where

$$I_m = \int_0^1 (1 - x^m)^{-1/m} dx \qquad (2.2.21)$$

I_m can only be evaluated using numerical integration; thus we can only obtain an approximation to f(m). A suitable function to fit is:

$$f(m) \approx 1 - \frac{1}{q(m)} \qquad q(m) \in P_2(m) \qquad (2.2.22)$$

By performing the integration over the range m = 2.0 (0.5) 6.0 and fitting q(m) by least squares, it was found that :

$$q(m) = 0.6095220m^2 - 0.0168260m + 0.3456630 \qquad (2.2.23)$$

gave $f(m) \equiv I_m = 1.0 \pm 0.0003$ within this range. (The maximum error occurring only at the extreme lower end of the range). Quadrature was performed by dividing the range (0,1) into 10,001 equal strips, using Simpson's Rule on all but the last one, then dividing this last strip into 10,000 substrips, using Simpson's Rule again, and the mid-point rule for the final substrip, since $(1 - x^m)^{-1/m} \to \infty$ as $x \to 1$. Graphs of the resulting curves $p_m(x)$ for m = 2 and m = 6 appear in fig.2.2.7.

Stress distributions in a finite soil layer

The effect of a rigid base or substratum on the stress distribution within the soil layer is a more difficult problem. Fortunately it has a lesser effect on soil displacements than the footing rigidity. There are two limiting cases which have been considered:

a. that of a smooth rigid base, and

b. where the soil adheres fully to the base (ie. a perfectly rough base).

The complete solution for stresses and displacements in a soil layer on a smooth rigid base, from a line-load at the soil surface, was given by Filon (1903). He first considered an elastic beam extending to infinity in the xy-plane, and lying between $z = -D$ and $z = D$, with a line-load of intensity P/unit length in the y-direction applied to the upper surface above the origin - see Fig.2.2.8. We suppose the beam to be supported on knife-edges sufficiently far away not to effect the stresses and displacements. By integrating the equations of equilibrium for the beam, using a stress function composed of trigonometric and hyperbolic functions, applying the stress boundary conditions and then expanding terms of the form sin u sinh u in power series form around the origin, he obtained (p.98) solutions for the stresses σ_x, σ_z, τ_{xz} at the point (x,z) in the beam in the form:

$$\sigma_x = \frac{P}{\pi D} \sum_{n=0}^{\infty} \left(\frac{r}{D}\right)^n \frac{\cos n\phi}{n!} \left[F_n - G_{n+1} + \frac{z}{D} F_{n+1}\right]$$

$$\sigma_z = \frac{P}{\pi D} \sum_{n=0}^{\infty} \left(\frac{r}{D}\right)^n \frac{\cos n\phi}{n!} \left[F_n + G_{n+1} - \frac{z}{D} F_{n+1}\right] \qquad (2.2.24)$$

$$\tau_{xz} = \frac{P}{\pi D} \sum_{n=1}^{\infty} \left(\frac{r}{D}\right)^n \frac{\sin n\phi}{n!} \left[G_{n+1} - \frac{z}{D} F_{n+1}\right]$$

Where:

$r = \sqrt{x^2 + z^2}$, $\cos \phi = z/r$ as in Fig.2.2.8 and the coefficients F_n and G_n are given by the infinite integrals:

$$F_1 = \int_0^\infty \left(\frac{\cosh u}{\sinh 2u - 2u} - \frac{3}{4u^2} \right) du$$

$$F_{2n+1} = \int_0^\infty \left(\frac{u^{2n+1} \cosh u}{\sinh 2u - 2u} \right) du \qquad n > 0$$

$$F_{2n} = \int_0^\infty \left(\frac{u^{2n} \sinh u}{\sinh 2u + 2u} \right) du \qquad n \geq 0 \quad (2.2.25)$$

$$G_{2n+1} = \int_0^\infty \left(\frac{u^{2n+1} \cosh u}{\sinh 2u + 2u} \right) du \qquad n \geq 0$$

$$G_{2n} = \int_0^\infty \left(\frac{u^{2n} \sinh u}{\sinh 2u - 2u} \right) du \qquad n > 0$$

These series are uniformly and absolutely convergent inside a circle radius D, centred at the origin. Filon found the approximate values of the first few coefficients F_n and G_n by quadrature; the writer has performed numerical integration using Simpson's rule on the MULTICS computer system to obtain the coefficients up to F_8, G_8, and these are given in Fig.2.2.9.

It is now possible to model the effect of a smooth rigid base inserted in the beam at depth D, by superimposing on the above stress formulae the stress distribution resulting from an equal and opposite line-load P' applied to the lower surface of the beam directly beneath P (Fig.2.2.8.). This will, by symmetry, cause there to be no vertical displacements and zero shear stresses along the line z = 0, which are the correct boundary conditions at a smooth rigid base.

Filon plotted a graph of vertical stress σ_z at the base, ie. where z =0, reproduced in Fig.2.2.10 with a graph of the stress predicted by Boussinesq theory superimposed, from which it is seen that directly under the line-load the stress is about 1.44 times as great as that given by the Boussinesq theory, while away from this point the stress dies out very rapidly, vanishing at a distance of about 1.35D from the z axis.

To evaluate these stresses, the stress formulae from the loads P and P' may be combined, since the stress at (x, z) from P' is the same, by symmetry, as that at (x, - z) from P. Such a combination gives the following formulae for the stresses at (x, z) in a soil layer of depth D underlain by a smooth rigid

base:

$$\sigma_x = \frac{2P}{\pi D} \sum_{n=0}^{\infty} \left(\frac{r}{D}\right)^n \frac{\cos n\phi}{n!} X_n$$

$$\sigma_z = \frac{2P}{\pi D} \sum_{n=0}^{\infty} \left(\frac{r}{D}\right)^n \frac{\cos n\phi}{n!} Z_n \qquad (2.2.26)$$

$$\tau_{xz} = \frac{2P}{\pi D} \sum_{n=0}^{\infty} \left(\frac{r}{D}\right)^n \frac{\sin n\phi}{n!} T_n$$

where:

$$X_n = \begin{cases} F_n - G_{n+1} & \text{if } n \text{ is even} \\ (z/D)F_{n+1} & \text{if } n \text{ is odd} \end{cases}$$

$$Z_n = \begin{cases} F_n + G_{n+1} & \text{if } n \text{ is even} \\ -(z/D)F_{n+1} & \text{if } n \text{ is odd} \end{cases}$$

$$T_n = \begin{cases} G_{n+1} & \text{if } n \text{ is even} \\ -(z/D)F_{n+1} & \text{if } n \text{ is odd} \end{cases}$$

These combined formulae involve only the even-numbered F coefficients and the odd-numbered G coefficients. The size of F_n and G_n increases with n at about the same rate as n!, so that if r is close to D the series in equations 2.2.26 will be very slowly convergent.

By the method of discretization into line-loads, the stresses and hence displacements for a strip-load on a finite layer may be modelled; this is not feasible, however, without a modern programmable calculator. By the time these became available, advanced mathematical techniques had brought treatment of more complex physical problems into range. Timoshenko and Goodier (1970), p.57, outline the solution for strip-loading using a similar stress function to Filon's and representing the loading by Fourier series. Pickett (1938) treated the line-load problem for a rough base using Fourier integrals, and subsequent work has dealt almost exclusively with the rough-base situation, which is of more practical relevance, though leading to considerably more complex formulae. Burmister (1956) presented the complete solution in three dimensions for a point-load; his formulae were

integrated numerically by Poulos (1967) to give influence tables for stresses and displacements under any type of surface loading.

Ueshita and Meyerhof (1968) have obtained influence factors for the vertical displacements of the edge of a uniform strip-load on a finite soil layer with specified Poisson's ratio, for the cases of a smooth and of a rough base. These were used by the writer to investigate the error caused by using the Boussinesq infinite-depth stress theory to predict displacements on a finite layer. The Boussinesq formulae for stress under the edge of a strip-load, equations 2.2.12 - 14, may be integrated analytically with respect to z, giving by elasticity theory the centreline displacement w of a strip load of intensity P, width 2a, on a soil layer of depth D as:

$$w = \frac{2P}{\pi}\,\frac{1+\nu}{E}\left[(1-2\nu)\,D\tan^{-1}(a/D) + (1-\nu)\,a\ln(1+D^2/a^2)\right] \qquad (2.2.28)$$

Curves of centreline displacement against layer depth for E = P = 1 were plotted for $\nu = 0.0$ and $\nu = 0.5$, and appear in Fig.2.2.11 together with the Ueshita and Meyerhof smooth-base curves. There is very little difference in the curves for $\nu = 0.0$, but for $\nu = 0.5$ the Boussinesq theory considerably underestimates the correct displacements (by almost 100% for very shallow layers). This discrepancy in the case $\nu = 0.5$ is still large (about 25%) even for very deep layers ($D/a = 10$), indicating that the statement of Terzaghi (1943, p.423) that "in the upper half of the layer the state of stress is practically identical with that in an elastic, semi-infinite deposit which is acted upon by the same load", is valid only for the vertical stress σ_z.

The writer has obtained good agreement with the Ueshita and Meyerhof curves by discretizing the strip-load into line-loads, and for each line-load using the Filon formulae 2.2.26 for points within the semicircle $r < 0.5D$, $z > 0$. Outside this region, the Boussinesq formula 2.2.9 was used to give the vertical stress σ_z, but the horizontal stress σ_x and shear stress τ_{xz} were taken to be zero except in a narrow strip near the surface, $0 < Z < 0.05D$, where the Boussinesq formulae 2.2.10 - 11

were used. The resulting curves are sketched in Fig.2.2.11.

A stress distribution subroutine called FILON using this theory has been included in the program FDTIM, and the user may select this or the Boussinesq theory subroutine BOUSS in his analysis.

General Comments

From the foregoing it may be seen that the classical Boussinesq theory is amenable to some empirical corrections which improve its ability to model the more important boundary conditions of our problem. Nevertheless, the approach becomes slightly ponderous in this way, and certain other boundary conditions such as rough boundary surfaces are ruled out by the method of calculating the displacements, as given in § 2.3.

A great advantage of the finite element approach, to be described in Chapter 4, is that the most general boundary conditions and problem geometry can be handled routinely. This may be seen in the work of, for example, Milovic et al (1970), who have published influence tables for stress distribution and surface displacements of a rough rigid loaded strip on a finite soil layer underlain by a rough rigid base.

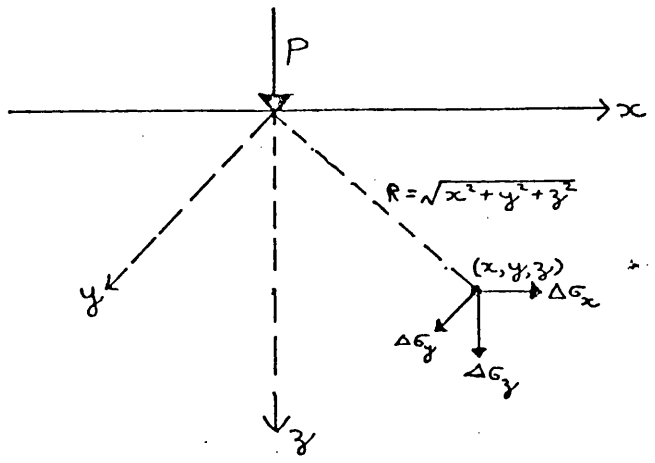


Fig. 2.2.1 The Boussinesq point-load problem

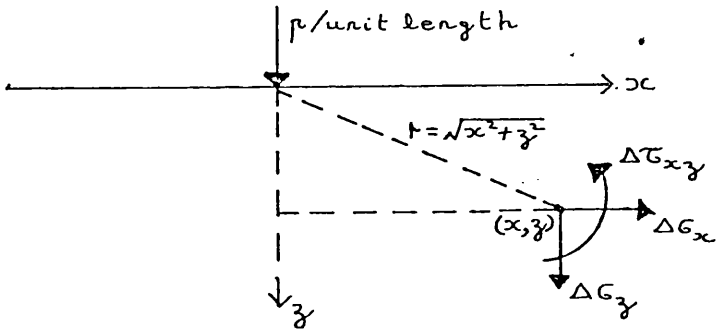


Fig. 2.2.2 Line-load plane strain problem

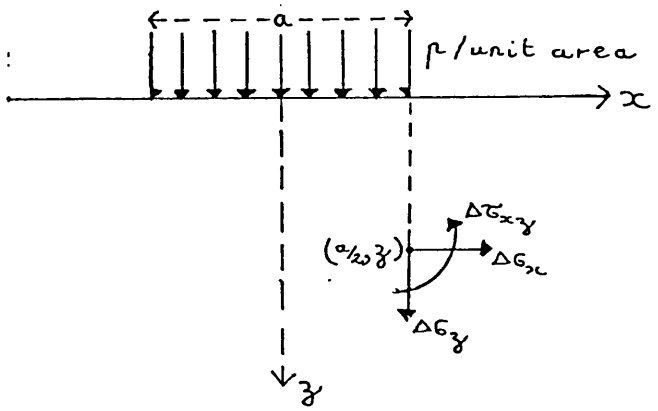


Fig. 2.2.3 Strip-load plane strain problem

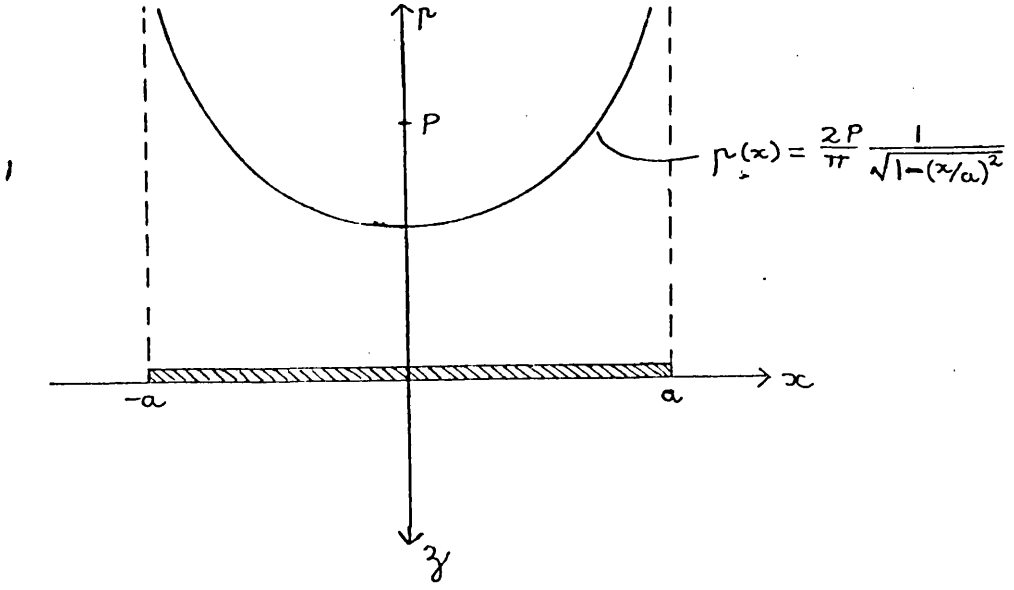


Fig. 2.2.4 Contact pressure across rigid footing on half-space

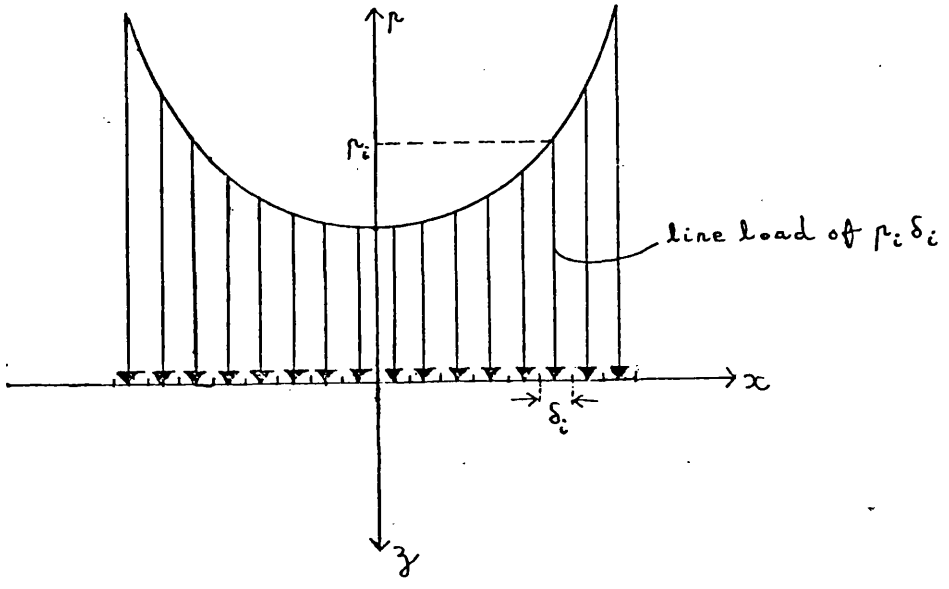


Fig. 2.2.5 Discretization into line-loads

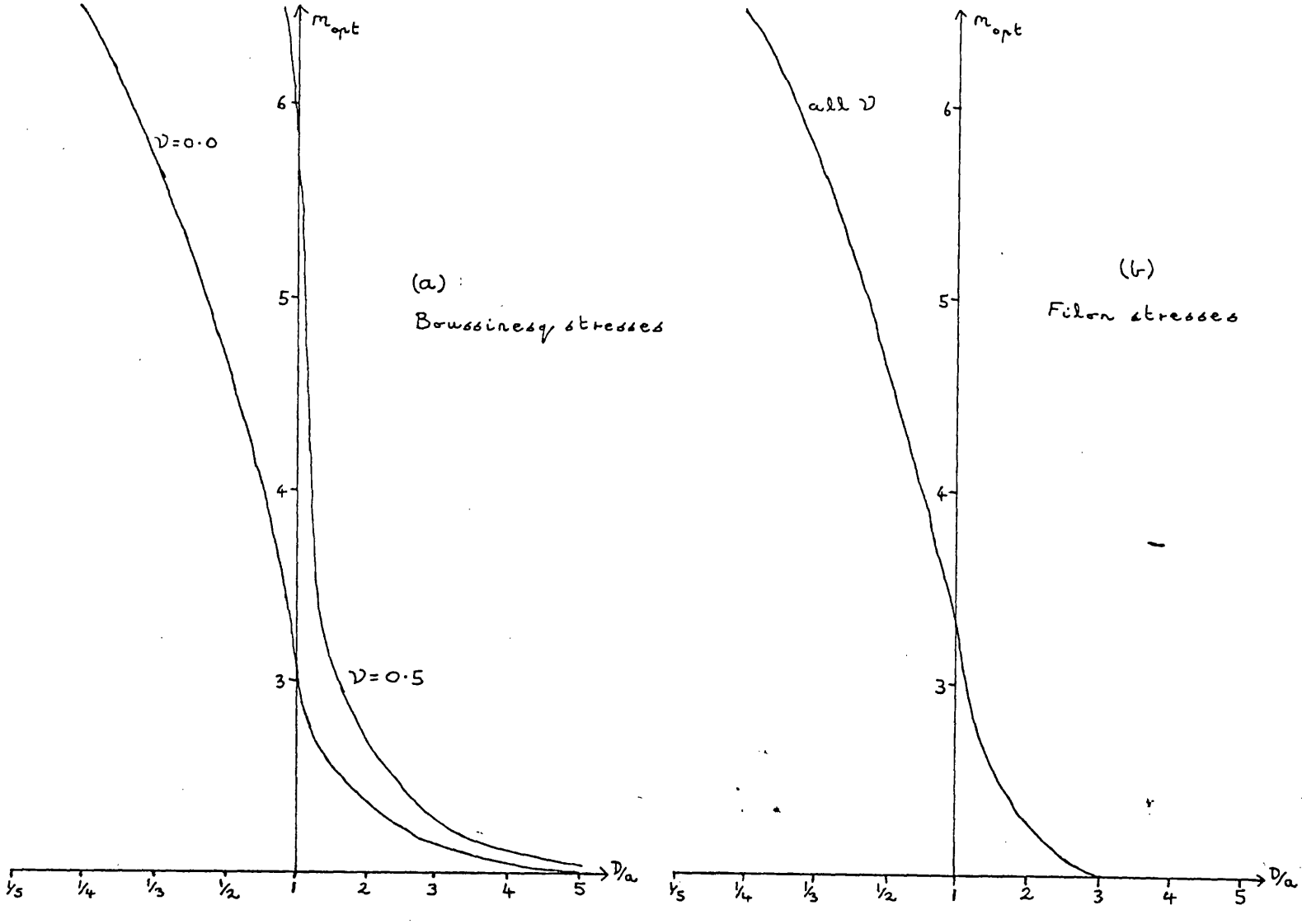
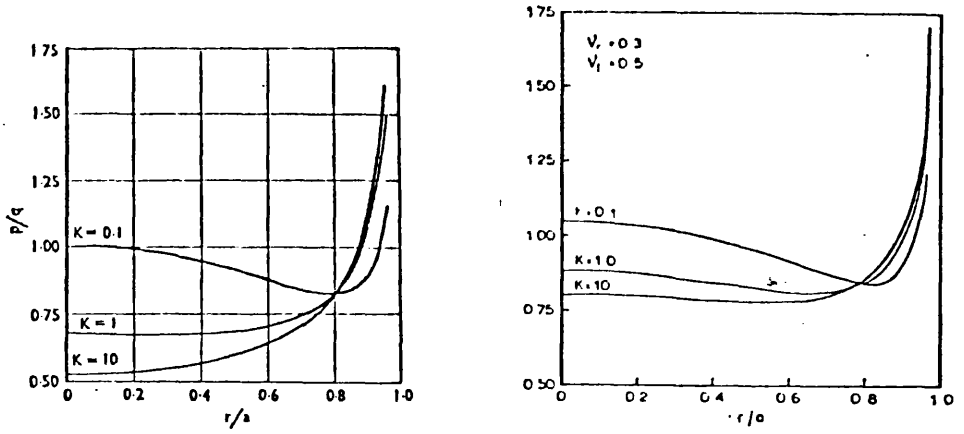


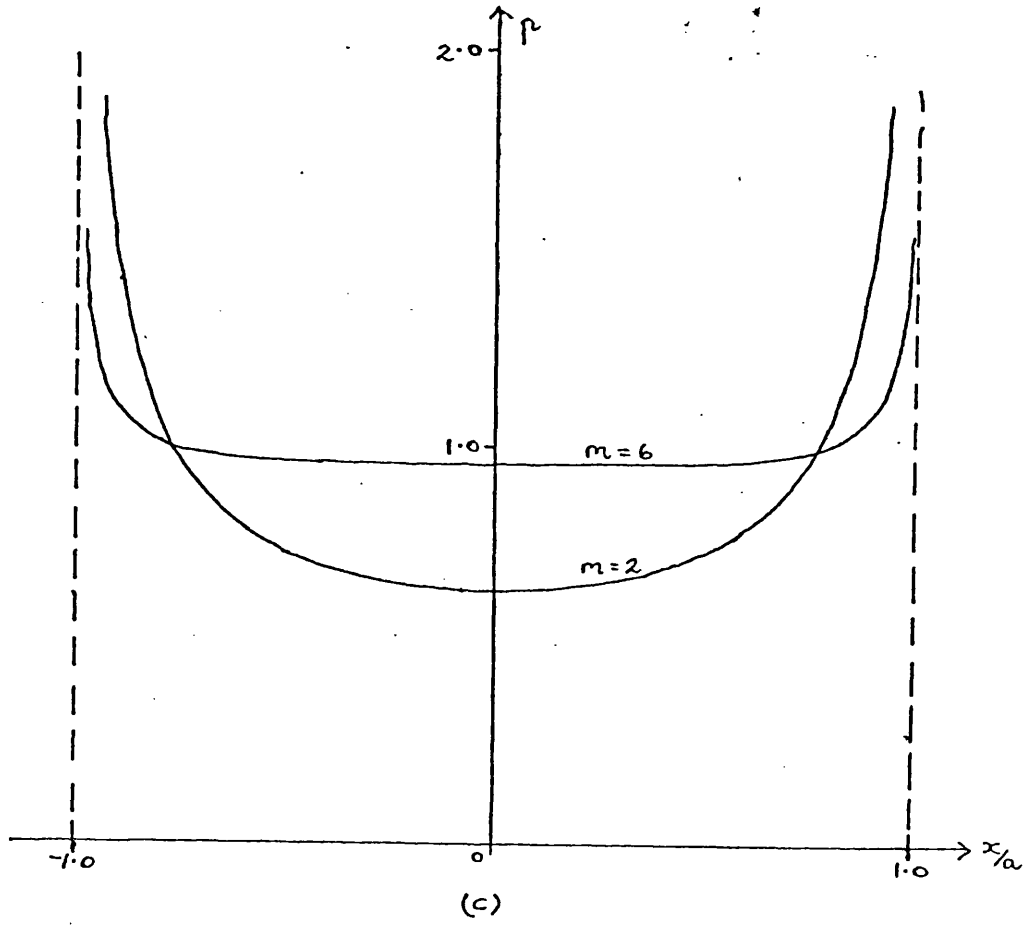
Fig. 2.2.6 Optimum values of m for finite-depth soil layers



(a) (b)

Deep foundations Shallow foundations

(after Brown (1969 a, b))



(c)

Using equation 2.2.19

Fig. 2.2.7 Contact pressure distributions

n	F_n	G_n
0	0·527	
1	0·438	0·9125
2	1·740	2·8042
3	7·210	5·750
4	23·38	24·825
5	122·0	119·0
6	717·0	723·0
7	5050	5033
8	40294	40344

Fig. 2.2.9 Values of F_n and G_n

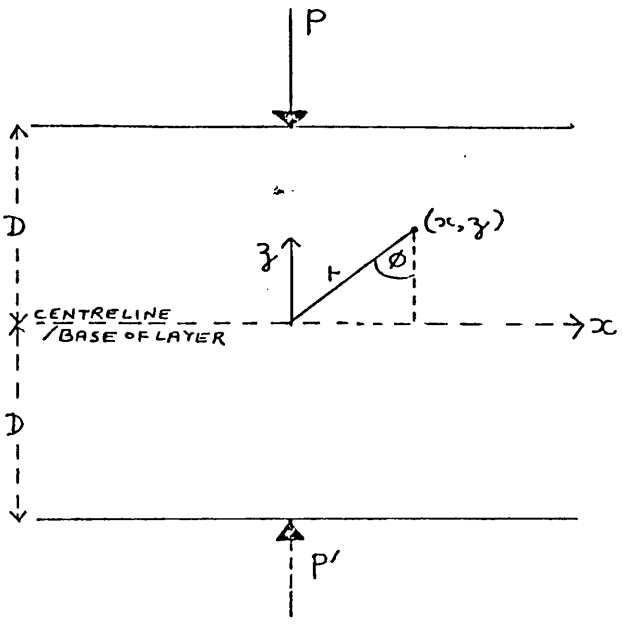


Fig. 2.2.8 The symmetric line-load problem

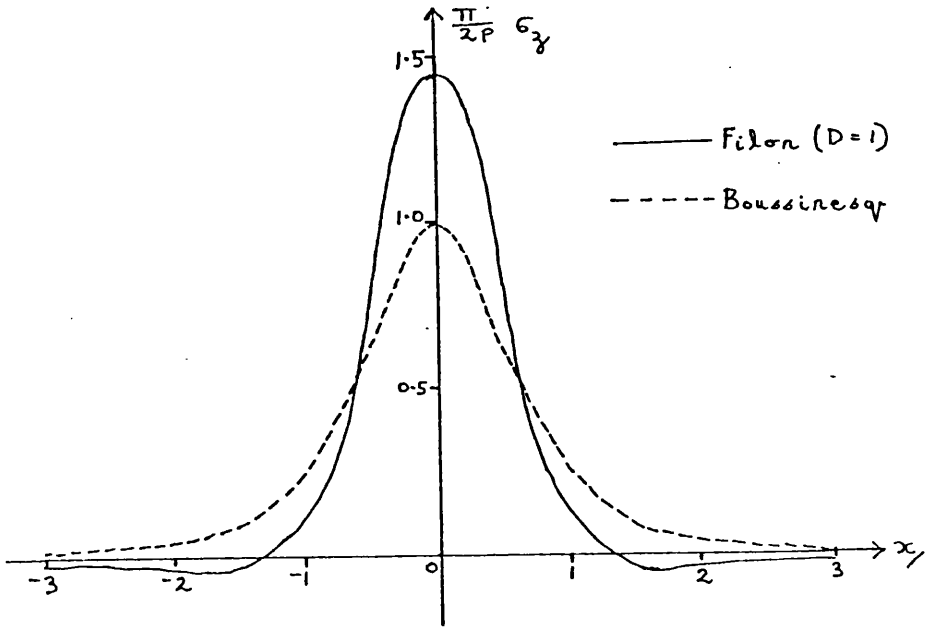


Fig. 2.2.10 Vertical stress on base of layer

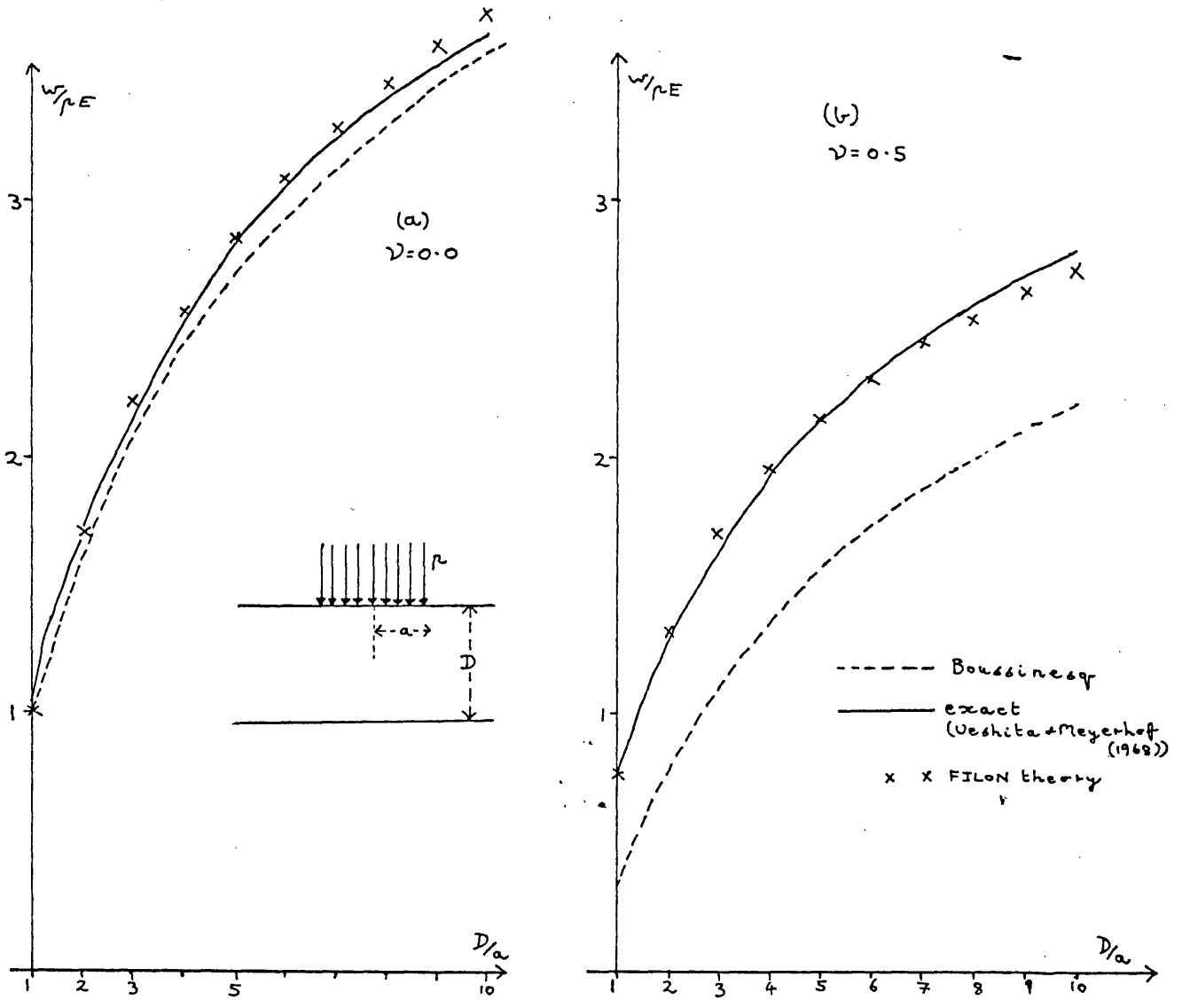


Fig. 2.2.11 Layer depth - centreline settlement curves

2.3 Elasticity Theory

In this section we summarize the fundamentals of elasticity theory, showing how they are used in the program FDTIM.

For a linear elastic isotropic body in three dimensions, direct stress increments $\Delta\sigma_x$, $\Delta\sigma_y$, $\Delta\sigma_z$ cause strains ε_x, ε_y, ε_z, given by the elasticity equations:

$$\varepsilon_x = \frac{1}{E}\left[\Delta\sigma_x - \nu(\Delta\sigma_y + \Delta\sigma_z)\right] \qquad (2.3.1)$$

$$\varepsilon_y = \frac{1}{E}\left[\Delta\sigma_y - \nu(\Delta\sigma_x + \Delta\sigma_z)\right] \qquad (2.3.2)$$

$$\varepsilon_z = \frac{1}{E}\left[\Delta\sigma_z - \nu(\Delta\sigma_y + \Delta\sigma_x)\right] \qquad (2.3.3)$$

For the case of plane strain, the symmetry in the out-of-plane y-direction imposes the condition $\varepsilon_y = 0$. This determines the stress increment $\Delta\sigma_y$ induced by the applied increments $\Delta\sigma_x$ and $\Delta\sigma_z$:

$$\Delta\sigma_y = \nu(\Delta\sigma_x + \Delta\sigma_z) \qquad (2.3.4)$$

The vertical strain ε_z, which is the quantity of primary interest, is then given by:

$$\varepsilon_z = \frac{1+\nu}{E}\left[(1-\nu)\,\Delta\sigma_z - \nu\Delta\sigma_x\right] \qquad (2.3.5)$$

Displacements may be found by integrating strains. For a soil layer of depth D in plane strain (Fig.2.3.1) the vertical displacement at a given point (X, Z) is

$$w(X, Z) = \int_Z^D \varepsilon_z(X, z)\, dz \qquad (2.3.6)$$

where $\varepsilon_z(x, z)$ is the vertical strain at the point (x, z).

Discretization

To model this problem, we discretize the soil mass into a mesh of points. The infinite extension in the x-direction must be approximated by choosing finite limits, say $x \in [A, B]$. Then the interval $[A, B]$ is partitioned into a finite set $\{x_1 = A, x_2, \ldots, x_{N_x} = B\}$ and the z-interval $[0, D]$ is

partitioned into $\{z_1 = 0, z_2, \ldots, z_{N_z} = D\}$, as in Fig.2.3.2
This gives us a set of mesh-points $\{(x_i, z_j) \mid i = 1, \ldots, N_x, j = 1, \ldots, N_z\}$.

Stress distribution

Stress increments may now be calculated at every mesh-point, using the method of approximating the footing load by a series of line-loads, as proposed in section 2.2. Suppose the footing lies between mesh-lines $x = x_\ell$ and $x = x_r$, and the contact pressure distribution is $p(x)$, $x_\ell < x < x_r$, as in Fig.2.3.2. Then we divide the footing into S equal intervals, say, of width $\delta = (x_r - x_\ell)/S$ and approximate the footing load by line-loads $P_1, \ldots, P_S$, where line-load P_K is at $z = 0$, $x = x_\ell + (K - \frac{1}{2})\delta$, and has magnitude $P_K = p(x_\ell + (K - \frac{1}{2})\delta).\ \delta$ (cf. Fig.2.2.5).

The stresses at the mesh-point (x_i, z_j) are then given from equations 2.2.9 - 11 by:

$$\Delta\sigma_{z_{ij}} = \frac{2}{\pi} \sum_{k=1}^{S} P_K \frac{Z_j^3}{R_{ijk}^4} \qquad (2.3.7)$$

$$\Delta\sigma_{x_{ij}} = \frac{2}{\pi} \sum_{k=1}^{S} P_K \frac{X_{ik}^2 Z_j}{R_{ijk}^4} \qquad (2.3.8)$$

$$\Delta\tau_{xz_{ij}} = \frac{2}{\pi} \sum_{k=1}^{S} P_K \frac{X_{ik} Z_j^2}{R_{ijk}^4} \qquad (2.3.9)$$

Where $X_{ik} = x_\ell + (k - \frac{1}{2})\delta - x_i$

$$Z_j = z_j - z_o$$

$$R_{ijk} = \sqrt{X_{ik}^2 + Z_j^2}$$

On the surface between x_ℓ and x_r the direct stresses are equal to the contact stress, ie.

$\Delta\sigma_{z_{il}} = \Delta\sigma_{x_{il}} = p(x_i)$, $\Delta\tau_{xz_{il}} = 0$ $i = \ell + 1, \ldots, r - 1$ and

away from the footing all stress increments are zero. At the singular points at the edges of the footing, we may take the stress increments to be zero, so that the surface displacements predicted will be for the soil surface rather than for the footing edge.

Strains and displacements

By the elasticity equation 2.3.5 we can now find the vertical strain at each mesh-point (x_i, z_j):

$$\varepsilon_{z_{ij}} = \frac{1 + \nu}{E}\left[(1 - \nu)\Delta\sigma_{z_{ij}} - \nu\Delta\sigma_{x_{ij}}\right] \qquad (2.3.10)$$

Displacements in the z-direction at any mesh-point are found by replacing the integral in equation (2.3.6) by numerical integration using the trapezium rule:

$$w(x_i, z_j) = \tfrac{1}{2}\sum_{k=j}^{N_z-1} (\varepsilon_{z_{i,k}} + \varepsilon_{z_{i,k+1}})(z_{k+1} - z_k) \qquad (2.3.11)$$

Principle of effective stress

The Boussinesq theory of section 2.2 provides total stress increments (ie. ignoring any induced change in the internal pore-water pressure of the soil). Use of these increments together with the drained values of Young's modulus and Poisson's ratio for the soil, E' and ν', will yield the drained or long-term soil displacements. Alternatively, use of the undrained parameters E_u and ν_u will give the undrained, or immediate, settlement. To perform an analysis for some intermediate time, while the process of consolidation is occuring, the drained soil parameters E' and ν' are used, together with the effective stress increments at that time, ie. the stress increments actually applied to the soil skeleton. Terzaghi (1923) first related total and effective stress in saturated soils by:

$$\sigma' = \sigma - u \qquad (2.3.12)$$

where u is the internal pore-water pressure in the soil.

Taking increments, we have:

$$\Delta\sigma' = \Delta\sigma - \Delta u \qquad (2.3.13)$$

so that a knowledge of the excess pore-water pressure distribution prevailing at any given time, is required in order to define the time pattern of effective stress. (Of course, the long-term settlement is reached when the excess pore-pressure has completely dissipated, so that $\Delta u = 0$ and $\Delta\sigma' = \Delta\sigma$).

We must thus investigate firstly the immediate excess pore-pressure caused by a total stress increase, and secondly (in Chapter 3) the dissipation of this pressure with time.

2.4 Undrained Excess Pore-Pressure Distribution

Considering a soil sample in a triaxial apparatus (where the non-major principal stresses are equal: $\sigma_2 = \sigma_3$), Skempton (1954) proposed that principal stress increments $\Delta\sigma_1$ and $\Delta\sigma_3$ applied to the soil cause an undrained pore-pressure increase of:

$$\Delta u = B\left[\Delta\sigma_3 + A(\Delta\sigma_1 - \Delta\sigma_3)\right] \qquad (2.4.1)$$

the parameters A and B being the Skempton pore-pressure coefficients. As equation 2.4.1 may in fact be derived from equations 2.3.1 - 3, it can readily be shown that for an ideal saturated elastic soil and incompressible pore-fluid, B = 1 and A = 1/3. It is admissable to consider that in all saturated soils the value of B remains close to unity; however, because of the very significant effects of stress history on the stress strain behaviour of soil, the parameter A must be measured experimentally (see Fig.2.4.1).

In 1958 Henkel generalised equation 2.4.1 to cover three dimensional stress situations,as:

$$\Delta u = 1/3\ (\Delta\sigma_1 + \Delta\sigma_2 + \Delta\sigma_3) + \alpha\sqrt{(\Delta\sigma_1 - \Delta\sigma_2)^2 + (\Delta\sigma_2 - \Delta\sigma_3)^2 + (\Delta\sigma_1 - \Delta\sigma_3)^2} \qquad (2.4.2)$$

with an experimental pore-pressure coefficient α. This formulation was later published by Skempton (1960). For the triaxial test situation, equation 2.4.2 reduces, using $\Delta\sigma_2 = \Delta\sigma_3$, to :

$$\Delta u = (1/3 + \alpha\sqrt{2})\ \Delta\sigma_1 + (2/3 - \alpha\sqrt{2})\ \Delta\sigma_3 \qquad (2.4.3)$$

so that, comparing coefficients with equation 2.4.1,

$$A = \alpha\sqrt{2} + 1/3 \qquad (2.4.4)$$

An ideally elastic soil has $\alpha = 0$.

From equation 2.4.1 we can establish a relationship between the drained and undrained moduli of soil. In triaxial conditions , the equation for axial strain, 2.3.3, becomes:

$$\varepsilon_1 = \frac{1}{E}\left[\Delta\sigma_1 - 2\nu\Delta\sigma_3\right] \qquad (2.4.5)$$

As mentioned before, the immediate axial strain can be determined either using total stress increments and the undrained soil parameters E_u, ν_u, or using effective stress increments and the soil skeleton parameters E' and ν'. Thus,

$$\varepsilon_z = \frac{1}{E_u}\left[\Delta\sigma_1 - 2\nu_u\Delta\sigma_3\right] = \frac{1}{E'}\left[\Delta\sigma_1 - 2\nu'\Delta\sigma_3'\right] \qquad (2.4.6)$$

and by substituting equations 2.3.13 and 2.4.1 and equating coefficients of $\Delta\sigma_1'$ and $\Delta\sigma_3'$, we find that $\nu_u = \frac{1}{2}$ and

$$E' = (1 - A + 2\nu'A)\, E_u \qquad (2.4.7)$$

For an ideally elastic soil, A = 1/3 and so $E' = 2/3\ (1 + \nu')E_u$ which agrees with the fact that the drained and undrained shear moduli are equal (since shear stress is unchanged by pore-pressure).

It is easy to forget in using this theory that the pore-pressure constant A, which can be determined in a triaxial test, is valid only for triaxial conditions. It can however, be shown that for a soil in plane strain equation 2.4.1 also holds, although the pore-pressure coefficient A is different from the triaxial case; we shall denote it Ap, and establish a linear relationship between A and Ap.

For an undrained soil in plane strain,

$$\Delta\sigma_y = \nu_u(\Delta\sigma_x + \Delta\sigma_z) = \tfrac{1}{2}(\Delta\sigma_x + \Delta\sigma_z) \qquad (2.4.8)$$

So as $\Delta\tau_{xy} = \Delta\tau_{zy} = 0$, we may take $\Delta\sigma_y = \Delta\sigma_2$; taking the x,z coordinates in the major and minor principal stress directions, equation 2.4.2 becomes:

$$\Delta_u = \tfrac{1}{2}(\Delta\sigma_1 + \Delta\sigma_3) + 3/2\ \ \alpha(\Delta\sigma_1 - \Delta\sigma_3)$$

$$= (\tfrac{1}{2} + \alpha\sqrt{3/2})\Delta\sigma_1 + (\tfrac{1}{2} - \alpha\sqrt{3/2})\Delta\sigma_3 \qquad (2.4.9)$$

Hence, the plane strain pore-pressure coefficient Ap is related to the three-dimensional coefficient α by:

$$Ap = \alpha\sqrt{3/2} + \tfrac{1}{2} \qquad (2.4.10)$$

and for an ideally elastic soil with $\alpha = 0$, we have $Ap = \tfrac{1}{2}$.

Eliminating α with equation 2.4.4, we find:

$$Ap = (3/2)A + \tfrac{1}{2}(1 - 1/\sqrt{3}) \qquad (2.4.11)$$

In our plane strain soil model, we will therefore establish the immediate excess pore-pressure at each mesh-point by finding the principal stress increases, using the relation

$$\left.\begin{matrix}\sigma_1 \\ \sigma_3\end{matrix}\right\} = \tfrac{1}{2}(\sigma_x + \sigma_z) \pm \tfrac{1}{2}\sqrt{(\sigma_x - \sigma_z)^2 + 4\tau_{xz}^2} \qquad (2.4.12)$$

and

$$\Delta u_{ij} = \Delta\sigma_{3_{ij}} + Ap(\Delta\sigma_{1_{ij}} - \Delta\sigma_{3_{ij}}) \qquad (2.4.13)$$

The pore-pressure coefficient A has been found to vary with the over-consolidation ratio, O.C.R., of the soil, as in Fig.2.4.1 (after Henkel (1960)). The relation between A and the plane strain coefficient Ap (equation 2.4.11) is graphed in Fig.2.4.2.

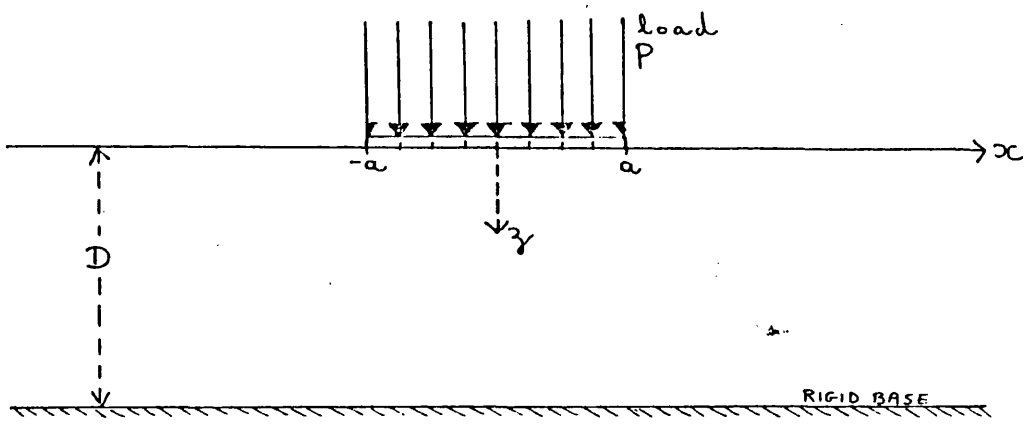


Fig. 2.3.1 Loaded soil layer in plane strain

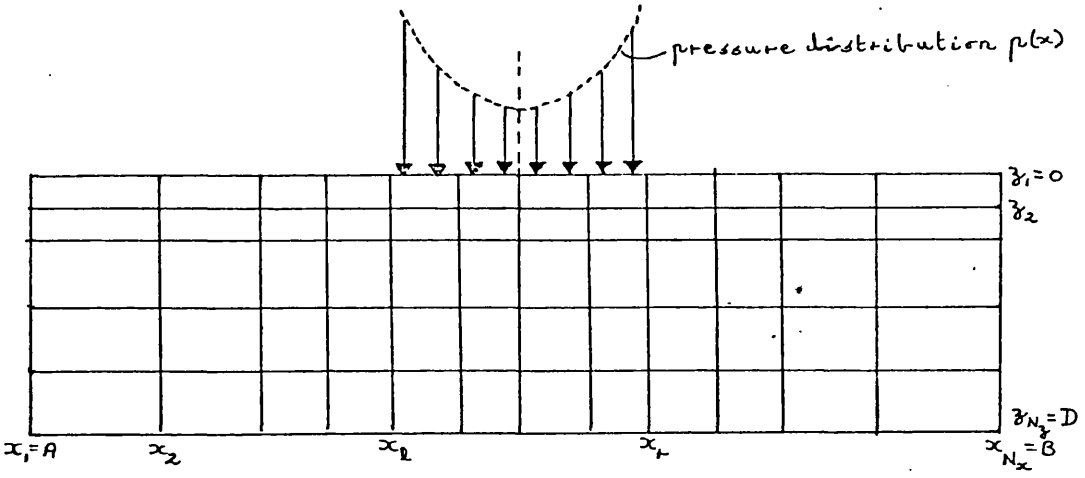


Fig. 2.3.2 Discretization of problem

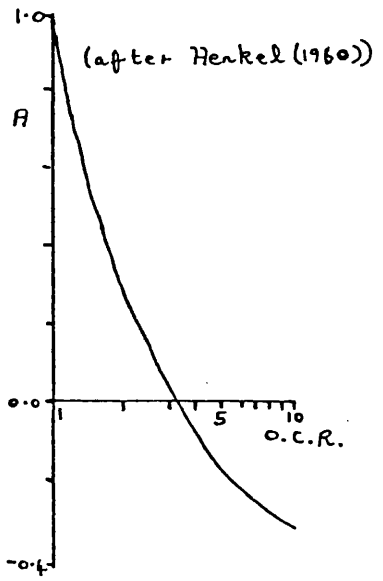


Fig. 2.4.1 Variation of A with overconsolidation ratio

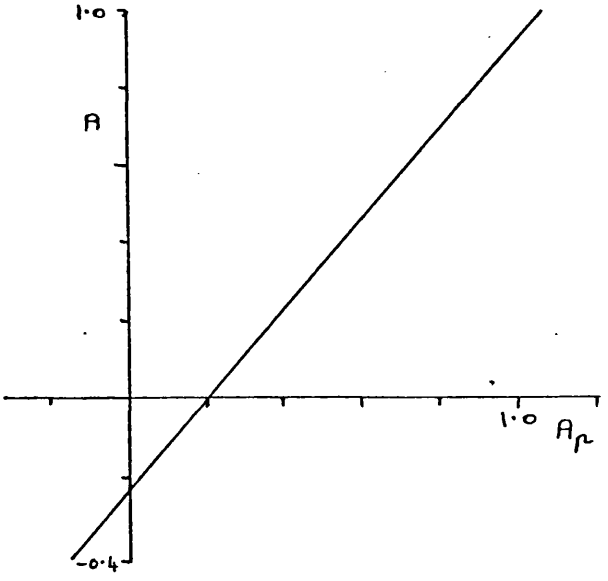


Fig. 2.4.2 Relation between A and A_p

2.5 Conclusions: Extensions and Restrictions

The theory has been developed for a numerical model which is capable of analysing displacements of a soil layer in plane strain supporting a rigid or flexible footing load. By the principle of superposition, the effect of further independent footings, loaded simultaneously, could also be modelled. (The extension to linked footings, where the loads depend on the relative settlements, is considered in § 3.4).

In physical terms, the elasticity equations 2.3.1 - 3 applied to the boundary mesh-points imply all boundaries to be smooth, since the effect of friction cannot be modelled, and in allowing horizontal displacements at the side boundaries $x = A$ and $x = B$ the model is consistent with an infinitely extending soil layer rather than one with rigid side walls. In this last respect the method has an advantage over the finite element method, where the side boundaries have to be taken far enough away not to influence the footing settlement, to model this more practically useful situation.

There is also the possibility of dealing with non-homogeneous soils, replacing the constant soil parameters E and ν by $E(x,z)$ and $\nu(x,z)$, evaluated separately at each mesh point. Non-linear stress strain laws (See Chapter 5) may be incorporated, using an incremental or an iterative procedure which require re-solution for the strains a number of times.

An incremental method would be needed also if the induced excess pore-pressure were taken to be dependent on shearing strains as well as direct stress increments, as proposed by Lo (1969). Wood (1972) has incorporated this relationship into Skempton's theory in § 2.4 by replacing the pore-pressure coefficient A by $A_t = A_t(\sigma_1,\sigma_3)$ using a hyperbolic pore-pressure axial strain model. However, as discussed fully in the Introduction, it is necessary to ask in each case if the increase in complexity and computing time needed to implement any adjustment to the linear elastic theory, is justified by a sufficient improvement in accuracy of modelling of a range of practical problems using field and laboratory data.

CHAPTER 3 : Theory of Consolidation, and the finite difference program FDTIM

3.1 Introduction

There are two further topics which need to be considered before a complete numerical model of the time-settlement behaviour of a frame structure on a soil layer can be constructed. These are firstly a theory of consolidation, and secondly a method of representing the structure. §3.2 develops and discusses the Biot and the Terzaghi-Rendulic theories of consolidation, and justifies the adoption of the simpler, uncoupled Terzaghi-Rendulic theory. In §3.3 we show how the governing equations may be approximated numerically using finite differences, and we apply the alternating direction implicit method of Peaceman and Rachford (1955) to the problem.

A flexibility matrix to represent the frame structure is developed in §3.4, and we conclude in §3.5 by assembling the methods described in this and the previous chapter to form the rectangular-mesh, finite-difference program FDTIM.

3.2 Theory of Consolidation

The differential equation governing the change of pore-water pressure in a soil with time was first deduced by Biot (1935). For an elastic saturated soil element composed of incompressible soil particles, the rate of pore-water expulsion must equal the rate of volume decrease, so that assuming the pore-water to be incompressible and to follow Darcy's law, we have

$$\frac{k_x}{\gamma_w}\frac{\partial^2 u}{\partial x^2} + \frac{k_y}{\gamma_w}\frac{\partial^2 u}{\partial y^2} + \frac{k_z}{\gamma_w}\frac{\partial^2 u}{\partial z^2} = -\frac{\partial \varepsilon_v}{\partial t} \qquad (3.2.1)$$

where k_x, k_y, k_z are the soil permeabilities in the x, y, z directions;

γ_w is the water density;

$u = u(x, y, z, t)$ is the excess pore-pressure at the point (x, y, z) at time t

and ε_v is the volumetric strain: $\varepsilon_v = \varepsilon_x + \varepsilon_y + \varepsilon_z$.

For the case of plane-strain, application of the elasticity equations 2.3.1-4 gives

$$\varepsilon_v = \frac{(1+\nu)(1-2\nu)}{E}\{\Delta\sigma_x + \Delta\sigma_z - 2u\} \qquad (3.2.2)$$

and substitution into 3.2.1 yields the differential equation

$$\frac{\partial u}{\partial t} = C_{v_x}\frac{\partial^2 u}{\partial x^2} + C_{v_z}\frac{\partial^2 u}{\partial z^2} + \frac{1}{2}\frac{\partial}{\partial t}(\sigma_x + \sigma_z') \qquad (3.2.3)$$

where

$$C_{v_x} = \frac{k_x}{\gamma_w}\frac{E}{2(1-2\nu)(1+\nu)}$$

and

$$C_{v_z} = \frac{k_z}{\gamma_w}\frac{E}{2(1-2\nu)(1+\nu)}$$

are the coefficients of consolidation in the x, z directions. Similar equations may be deduced for one- and three-dimensional problems (see Davis and Poulos (1972)).

Equation 3.2.3 embodies the 'true' or Biot two-dimensional consolidation theory. It may be simplified into a normal diffusion equation, however, by the assumption that the consolidation process does not cause any change in the bulk total stress $\frac{\sigma_x + \sigma_z}{2}$, ie there is no redistribution of total stresses as the footing settles. This removes the last term in the equation, leaving

$$\frac{\partial u}{\partial t} = C_{v_x}\frac{\partial^2 u}{\partial x^2} + C_{v_z}\frac{\partial^2 u}{\partial z^2}. \qquad (3.2.4)$$

giving a 'pseudo' two-dimensional consolidation theory, sometimes called the Terzaghi-Rendulic theory.

The main difference in predictions from the two theories is that the Biot theory can and does predict the Mandel-Cryer effect, which is impossible in a simple diffusion process. The Mandel-Cryer effect in its simplest form is that when a sphere of saturated soil is subjected to an all-round pressure, with free drainage at the surface, the pore-pressure in the centre is found to rise above its initial value during consolidation

before decaying. This effect was deduced from elasticity theory and Biot theory by Mandel (1953) and Cryer (1963), but in practical experiments it has proved rather more elusive than the Biot theory would predict. Davis and Poulos (1972) state that the Mandel-Cryer effect has been verified experimentally by Gibson, Knight and Taylor (1963) and by De Jong and Verruijt (1965). However, the results published by De Jong and Verruijt show the Mandel-Cryer effect only in the theoretical curves of pore-pressure against time predicted from Biot theory but not in their experimental results, and Gibson et al failed to observe any appreciable effect during virgin consolidation. They did observe the effect during swelling and reconsolidation cycles, and concluded that the Terzaghi-Rendulic theory is more applicable to consolidation in natural soil deposits. More importantly in the present context, Burland (1967) conducted model footing tests and found no appreciable change in the distribution of total stresses in the soil during consolidation - a condition which would validate the Terzaghi-Rendulic theory.

It therefore appears justified to use the Terzaghi-Rendulic 'pseudo' consolidation equation 3.2.4, which is amenable to a numerical method of solution as detailed in the following section, rather than the 'true' Biot theory. We shall see later that the Biot theory is, however, used in the finite element solution of the consolidation problem.

3.3 Finite Difference Solution

The explicit method of solution of equation 3.2.4, is obtained by making the approximations

$$\frac{\partial u}{\partial t} \simeq \frac{\Delta U_{i,j,n}}{\Delta t} = (U_{i,j,n+1} - U_{i,j,n})/\Delta t$$

$$\frac{\partial^2 u}{\partial x^2} \simeq \frac{\delta_x^2 U_{i,j,n}}{(\Delta x)^2} = (U_{i-1,j,n} - 2U_{i,j,n} + U_{i+1,j,n})/(\Delta x)^2 \quad (3.3.1)$$

$$\frac{\partial^2 u}{\partial z^2} \simeq \frac{\delta_z^2 U_{i,j,n}}{(\Delta z)^2} = (U_{i,j-1,n} - 2U_{i,j,n} + U_{i,j+1,n})/(\Delta z)^2$$

where $U_{i,j,n}$ is the approximate pore-pressure at the mesh point (x_i, z_j) at time t_n; it is inefficient because of the stability requirement

$$\left\{\frac{C_{v_x}}{(\Delta x)^2} + \frac{C_{v_z}}{(\Delta z)^2}\right\}\Delta t < \tfrac{1}{2} \qquad (3.3.2)$$

which puts a severe restriction on the maximum size of the timestep Δt, even for large times when the consolidation is extremely slow-moving.

This restriction would not apply to the Crank-Nicholson method:

$$\frac{\Delta U_{i,j,n}}{\Delta t} = \frac{1}{2}\Big\{C_{v_x}(\delta_x^2 U_{i,j,n} + \delta_x^2 U_{i,j,n+1})/(\Delta x)^2 + C_{v_z}(\delta_z^2 U_{i,j,n} + \delta_z^2 U_{i,j,n+1})/(\Delta z)^2\Big\} \qquad (3.3.3)$$

but with this method, for the mesh $\left\{(x_i, y_j)\,\middle|\, i=1,..,N_x;\ j=1,..,N_z\right\}$ we would have at least $(N_x - 2)(N_z - 2)$ equations to solve, each equation having five unknowns; this would be extremely laborious and expensive for all but the simplest meshes.

Fortunately, there exists a third method of much greater efficiency and with unconditional stability. It was proposed by Peaceman and Rachford (1955), and is known as the Alternating Direction Implicit (A.D.I.) method. It consists of separating the time-step into two equal half-steps. Over the first half-step we use a set of equations implicit in one of the directions, eg setting $\Delta t = t_{n+\frac{1}{2}} - t_n$,

$$(U_{i,j,n+\frac{1}{2}} - U_{i,j,n})/\Delta t = C_{v_x}(U_{i-1,j,n+\frac{1}{2}} - 2U_{i,j,n+\frac{1}{2}} + U_{i+1,j,n+\frac{1}{2}})/(\Delta x)^2 + C_{v_z}(U_{i,j-1,n} - 2U_{i,j,n} + U_{i,j+1,n})/(\Delta z)^2 \qquad (3.3.4)$$

and over the second half-step we make the method implicit in the other direction, viz

$$(U_{i,j,n+1} - U_{i,j,n+\frac{1}{2}})/\Delta t = C_{v_x}(U_{i-1,j,n+\frac{1}{2}} - 2U_{i,j,n+\frac{1}{2}} + U_{i+1,j,n+\frac{1}{2}})/(\Delta x)^2 + C_{v_z}(U_{i,j-1,n+1} - 2U_{i,j,n+1} + U_{i,j+1,n+1})/(\Delta z)^2 \qquad (3.3.5)$$

This method is stable for all ratios of $\Delta t/(\Delta x)^2$ and $\Delta t/(\Delta z)^2$ (see Smith (1965)).

Rearranging 3.3.4 and 3.3.5 with the unknowns on the left-hand-sides, and writing $V_{i,j}$ for the intermediate set of pore-pressures $U_{i,j,n+\frac{1}{2}}$:

$$-r_x V_{i-1,j} + (1+2r_x)V_{i,j} - r_x V_{i+1,j} = r_z U_{i,j-1,n} + (1-2r_z)U_{i,j,n} + r_z U_{i,j+1,n} \qquad (3.3.6)$$

and

$$-r_z U_{i,j-1,n+1} + (1+2r_z)U_{i,j,n+1} - r_z U_{i,j+1,n+1} = r_x V_{i-1,j} + (1-2r_x)V_{i,j} + r_x V_{i+1,j} \qquad (3.3.7)$$

where $r_x = C_{v_x}\frac{\Delta t}{(\Delta x)^2}$ and $r_z = C_{v_z}\frac{\Delta t}{(\Delta z)^2}$.

From this it can be seen that to progress from the n'th to the (n + 1)'th timestep we must solve two sets of tridiagonal equations, one set of order $N_z - 2$ and the other of order $N_x - 2$. It has been shown (Peaceman and Rachford (1955)) that for a typical problem the work involved in this is about seven times less than in the Crank-Nicholson method, and twenty five times less than in the explicit method with a stable timestep.

Generalization

The above theory has assumed constant mesh sizes Δx and Δz, and a homogeneous soil where C_{v_x} and C_{v_z} remain constant throughout. It is straightforward to allow for the mesh size to vary, getting coarser away from the loaded area, and doing this goes a long way to removing the objection raised by Wood (1972) that two-dimensional consolidation theory is too expensive to program because of the large number of mesh-points needed outside the immediate area of interest. The stability of the method is not affected.

A more interesting question is the generalization to layered soils, where we require the form of equations 3.3.6-7 applicable to the interface between two horizontal layers of different permeabilities and compressibilities. Abbott (1960) attempted to solve this by a method using fictitious hydraulic gradients, but his solution was later found to be inconsistent by Raymond

and Chan (1966). Murray (1971) produced a solution by arguing from first principles; this is avoided in the following solution, proposed by the writer.

Consider the one-dimensional flow problem, with a soil interface at mesh-point z_j, as in fig 3.3.1. We introduce the soil compressibility M_v, defined as

$$M_v = \frac{2}{E}(1-2\nu)(1+\nu) \qquad (3.3.8)$$

for plane strain, so that $C_v = \frac{k}{\gamma_w M_v}$. Suppose the upper soil layer (the 'p'-layer) has compressibility and permeability M_{v_p} and k_{z_p}, and the lower soil layer (the 'q'-layer) has corresponding parameters M_{v_q} and k_{z_q}.

There must be a unique value for the rate of flow of pore-water across the interface at any time, and this can be expressed in two ways by applying Darcy's Law to the 'p' and the 'q' layers:

$$v_j = \frac{k_{z_p}}{\gamma_w}\left(\frac{\partial u}{\partial z}\right)_{pj} = \frac{k_{z_q}}{\gamma_w}\left(\frac{\partial u}{\partial z}\right)_{qj} \qquad (3.3.9)$$

where $\left(\frac{\partial u}{\partial z}\right)_{pj}$ denotes $\frac{\partial u}{\partial z}$ evaluated at z_j approached from the 'p'-layer

and $\left(\frac{\partial u}{\partial z}\right)_{qj}$ denotes $\frac{\partial u}{\partial z}$ at z_j approached from the 'q'-layer.

(That these two gradients differ may be seen from the graph in fig 3.3.1.)

This fact leads to the continuity of flow equation

$$k_{z_p}\left(\frac{\partial u}{\partial z}\right)_{pj} = k_{z_q}\left(\frac{\partial u}{\partial z}\right)_{qj} \qquad (3.3.10)$$

We now apply the one-dimensional form of the consolidation equation 3.2.4 to the 'p'-layer at z_j and make a backward difference approximation for the second derivative

$$\left(\frac{\partial u}{\partial t}\right)_j = \frac{k_{z_p}}{\gamma_w M_{v_p}}\left(\frac{\partial^2 u}{\partial z^2}\right)_{pj} \simeq \frac{k_{z_p}}{\gamma_w M_{v_p}}\left\{\left(\frac{\partial u}{\partial z}\right)_{pj} - \left(\frac{\partial u}{\partial z}\right)_{p}\right\} / \tfrac{1}{2}\Delta z_j \qquad (3.3.11)$$

where $(\frac{\partial u}{\partial z})_p$ is $\frac{\partial u}{\partial z}$ evaluated at a point P midway between z_{j-1} and z_j.

Similarly for the 'q'-layer:

$$(\frac{\partial u}{\partial t})_j = \frac{k_{z_q}}{\gamma_w M_{v_q}}(\frac{\partial^2 u}{\partial z^2})_{qj} \simeq \frac{k_{z_q}}{\gamma_w M_{v_q}}\left\{(\frac{\partial u}{\partial z})_Q - (\frac{\partial u}{\partial z})_{qj}\right\} // \tfrac{1}{2}\Delta z_{j+1} \tag{3.3.12}$$

where Q lies midway between z_j and z_{j+1}.

Rearranging, we have

$$k_{z_p}(\frac{\partial u}{\partial z})_{pj} \simeq \tfrac{1}{2}M_{v_p}\gamma_w\Delta z_j(\frac{\partial u}{\partial t})_j + k_{z_p}(\frac{\partial u}{\partial z})_p \tag{3.3.13}$$

and

$$k_{z_q}(\frac{\partial u}{\partial z})_{qj} \simeq -\tfrac{1}{2}M_{v_q}\gamma_w\Delta z_{j+1}(\frac{\partial u}{\partial t})_j + k_{z_q}(\frac{\partial u}{\partial z})_Q \tag{3.3.14}$$

The left-hand-sides of these equations are equal, by the continuity of flow equation 3.3.10. Using this, we may solve for $(\frac{\partial u}{\partial t})_j$, giving

$$(\frac{\partial u}{\partial t})_j \simeq \frac{2\left\{k_{z_q}(\frac{\partial u}{\partial z})_Q - k_{z_p}(\frac{\partial u}{\partial z})_p\right\}}{\gamma_w(M_{v_p}\Delta z_j + M_{v_q}\Delta z_{j+1})} \tag{3.3.15}$$

But we may approximate the derivatives at P and Q by central differences, using the mesh-point values:

$$(\frac{\partial u}{\partial t})_j \simeq \frac{2\left\{k_{z_q}\left(\frac{U_{j+1} - U_j}{\Delta z_{j+1}}\right) - k_{z_p}\left(\frac{U_j - U_{j-1}}{\Delta z_j}\right)\right\}}{\gamma_w(M_{v_p}\Delta z_j + M_{v_q}\Delta z_{j+1})} \tag{3.3.16}$$

The right hand side of this equation thus gives a replacement for the term $C_{v_z}\frac{\partial^2 u}{\partial z^2}$ at the interface, and it reduces to this in the case of $k_{z_p} = k_{z_q}$ and $M_{v_p} = M_{v_q}$.

For the two-dimensional flow situation, we use this replacement in the consolidation equation 3.2.4, and for the term $C_{v_x} \frac{\partial^2 u}{\partial x^2}$ we use the normal finite difference representations of equations 3.3.4-5 with the soil parameters

$M_{v_j} = \frac{1}{2}(M_{v_p} + M_{v_q})$, $k_{x_j} = \frac{1}{2}(k_{x_p} + k_{x_q})$ and

$k_{z_j} = \frac{1}{2}(k_{z_p} + k_{z_q})$.

Equations 3.3.4-5 were written assuming uniform mesh-sizes Δx and Δz. It is straightforward to generalize them to the case of varying mesh-sizes, and by doing this and applying the adaptation for interfaces we obtain the most general form of the finite difference scheme, which is that used in the program FDTIM. The difference equations 3.3.4-5 become

$$-RX_1V_{i-1,j} + (1+2R)V_{i,j} - RX_2V_{i+1,j} = PZ_1U_{i,j-1} + \ldots$$
$$\ldots + (1-P-Q)U_{i,j} + QZ_2U_{i,j+1} \qquad (3.3.17)$$

and

$$-PZ_1U_{i,j-1} + (1+P+Q)U_{i,j} - QZ_2U_{i,j+1} = RX_1V_{i-1,j} + \ldots$$
$$\ldots + (1-2R)V_{i,j} + RX_2V_{i+1,j} \qquad (3.3.18)$$

where

$$X_1 = \frac{2\Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}}, \quad X_2 = \frac{2\Delta x_i}{\Delta x_i + \Delta x_{i+1}}$$

$$Z_1 = \frac{2\Delta z_{j+1}}{\Delta z_j + \Delta z_{j+1}}, \quad Z_2 = \frac{2\Delta z_j}{\Delta z_j + \Delta z_{j+1}}$$

$$P = \frac{2k_{z_{j-1}}.\Delta t}{\gamma_w.\Delta z_j\{M_{v_{j-1}}\Delta z_j + M_{v_{j+1}}\Delta z_{j+1}\}}$$

$$Q = \frac{2k_{z_{j+1}}.\Delta t}{\gamma_w.\Delta z_{j+1}\{M_{v_{j-1}}\Delta z_j + M_{v_{j+1}}\Delta z_{j+1}\}}$$

and

$$R = \frac{k_{x_j}.\Delta t}{\gamma_w M_{v_j}\Delta x_{i+1}\Delta x_i} \quad \text{where} \quad \Delta x_i = x_i - x_{i-1}, \text{ etc}$$

and the left-hand-sides of each equation are the updated variables for the next half-timestep.

Boundary Conditions

Soil boundaries fall into two types, permeable and impermeable. At a perfectly permeable boundary no excess pore-pressure can develop, and so if for example the base mesh-line $z = z_{N_z}$ were a permeable boundary we would set

$U_{i,N_z} \pm V_{i,N_z} = 0$ for all timesteps (Dirichlet boundary condition).

At an impermeable boundary, the flow across the boundary is zero, so that for an impermeable base we would use equation 3.3.18 for $j = N_z$ in the form

$$-(PZ_1 + QZ_2)U_{i,N_z-1} + (1 + P + Q)U_{i,N_z} = RX_1V_{i-1,N_z} + \ldots$$
$$\ldots + (1 - 2R)V_{i,N_z} + RX_2V_{i+1,N_z} \qquad (3.3.19)$$

obtained by setting $U_{i,N_z+1} = U_{i,N_z-1}$. In evaluating Q and R we use, by symmetry, $M_{v_{N_z+1}} = M_{v_{N_z-1}}$, $\Delta z_{N_z+1} = \Delta z_{N_z}$, etc.

Stability and Convergence of the A.D.I. Method

It was stated earlier that the A.D.I. method applied to the two-dimensional diffusion equation 3.2.4 is convergent and unconditionally stable. This is proved in Forsythe and Wasow (1959) using a Fourier series method. Problems of convergence have been encountered at soil layer boundaries, however, for large timesteps and where the distance from the boundary to the next parallel mesh-line is small.

Solution of the matrix equations

The tridiagonal matrix equations resulting from equations 3.3.17-18 were solved by the subroutine TRIDG1 written by Bloss (1976), which uses a variant of Gaussian elimination based on the Thomas algorithm.

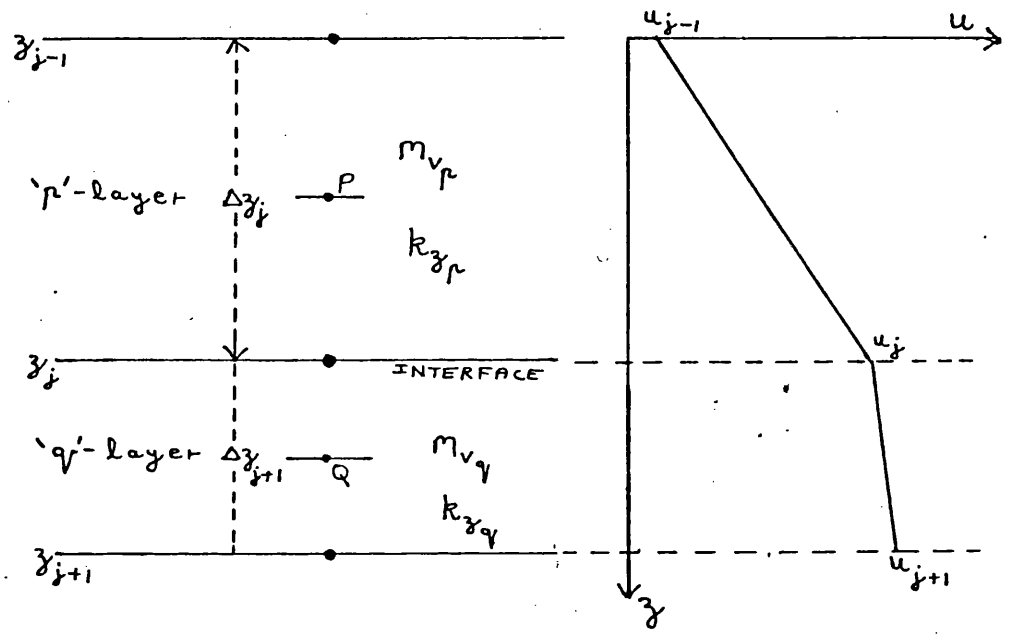


Fig. 3.3.1 Interface in finite difference mesh

3.4 Modelling the Structure

Using the theory of chapter 2 for finding stress and pore-pressure increment distributions, together with the finite difference approximation of the consolidation equation described in the previous section, we can model the settlement with time of a soil horizon under and around a loaded footing. By the principle of superposition, multiple footings independently loaded at varying times may also be considered. Our intention, however, is to be able also to model the problem of footings linked by some framed structure, as illustrated in fig 3.4.1. Here, the load on each footing is not constant, but depends upon the relative settlement of the footing, the structural load being continuously redistributed across the footings throughout the consolidation process.

Previous work on the soil-structure interaction problem has been reviewed in the introduction. Certain standard types of structure may be modelled using conventional structural analysis techniques (eg Chamecki (1956)), but in recent times the finite element method has proved a very powerful tool capable of completely analysing a wide range of structures. The only drawback of the method is the large amounts of computer store and time needed. This is particularly important in a consolidation analysis, where the stiffness equation must be solved one or more times at each timestep. In line with the idea of making FDTIM a small efficient program requiring limited storage, a procedure has been developed for rapidly forming the flexibility matrix for a more restricted class of structure.

Meyerhof (1953) proposed a way of approximating the framed structure of fig 3.4.1 by a single beam of equivalent stiffness. If the length of the frame is L, divided up into a number of bays of approximately equal length ℓ, then for each beam-level j the effective stiffness of the beam is augmented by the stiffness of the columns connecting adjacent beam-levels. The equivalent Moment of Inertia of the beam, I_j', he gives as

$$I_j' = I_j\left\{1 + \frac{K_\ell + K_u}{K_j + K_\ell + K_u} \cdot \frac{L^2}{\ell^2}\right\} \qquad (3.4.1)$$

where

$K_j = EI_j/\ell$ = average stiffness of beam at level j

$K_\ell = EI_\ell/h_\ell$ = average stiffness of lower columns

$K_u = EI_u/h_u$ = average stiffness of upper columns

h_ℓ = height of lower columns

h_u = height of upper columns.

Then the equivalent stiffness of the structure is $K' = \frac{EI'}{L}$

where

$$I' = \sum_j I'_j \qquad (3.4.2)$$

Meyerhof states that in a typical example this approximation predicted deformations correct to within about 5%. He also showed how to include the effect of cladding and load-bearing walls. It was subsequently used by Sommer (1965) to analyse a framed structure resting on a continuous strip footing (a raft). The method is of practical relevance since the deformations of primary importance to the engineer are the settlements of the footings, and specifically the differential settlements between adjacent footings. Skempton and McDonald (1956) have prepared tables of maximum allowable differential settlement in buildings. Litton and Buston (1968), looking purely at the effect on a framed structure of specified differential settlements of its footings, have shown that the greatest stresses and deformations are developed in the structural elements at the base of the building.

We therefore intend to confine our attention to a single-storey multi-bay portal, as in fig 3.4.2. As described above, a multi-storey building can be approximated by such a portal in a straightforward manner. We will allow the bays to have different lengths, and different stiffnesses along the beam element, thus enabling variations in structure or number of storeys across the building to be modelled. The columns from each footing are

assumed to be pin-jointed to the beam; this is acceptable if their stiffness has been taken into account already in calculating the equivalent beam stiffness.

Let there be M footings (ie M-1 bays), and let the i'th bay have half-length ℓ_i and beam inertia I_i, as in fig 3.4.2. Dead loads from the superstructure (if any) will be transmitted via the columns to the ends of each bay, while loads resting on the beam, and the weight of the beam itself, may be approximated by loads lumped at the ends and the mid-point of each bay. We will therefore consider loads applied in the positions shown in fig 3.4.2. The problem to be solved can then be expressed as follows:

If the external superimposed loads applied to the beam are established and the footings are vertically displaced by prescribed amounts, we would wish to then calculate the reactions at each footing and the beam displacements in the mid-points of each bay.

This forms an essential preliminary to the time-dependent problem, where footing displacements are occasioned by soil compression, and adjust themselves interactively as the footing reactions immediately change.

We start with the equations of equilibrium for a bar, length ℓ, Young's modulus E, inertia I, subjected to forces F_A, F_B and moments M_A, M_B at its end-points (fig 3.4.3). Clearly, $F_A = F_B = F$, say, and for equilibrium we have the well-established solution

$$M_A = \frac{2EI}{\ell}\{2\theta_A + \theta_B + {}^3/_\ell(\delta_A - \delta_B)\} \qquad (3.4.3)$$

$$M_B = \frac{2EI}{\ell}\{\theta_A + 2\theta_B + {}^3/_\ell(\delta_A - \delta_B)\} \qquad (3.4.4)$$

and

$$F.\ell = M_A + M_B \qquad (3.4.5)$$

The derivation of these equations may be found in most structural analysis textbooks, for example Horne and Merchant (1965).

Let us divide our beam into a number of elements and node-points (the node-points separating the elements being at the ends and mid-points of the bays). Let element n be that part of the beam between node-points n and n+1 (we need not at this stage discriminate between node-points at bay ends and mid-points), with a stiffness $K_n = \frac{E_n I_n}{\ell_n}$ and with forces F_n and moments $M_{A,n}$, $M_{B,n}$ acting upon it as in fig 3.4.4.

Then there are four quantities of interest at each node-point:

(i) δ_n, the normal displacement

(ii) θ_n, the angle of rotation of the beam

(iii) Q_n, the shear force downwards: $Q_n = F_n - F_{n-1}$

and (iv) M_n, the moment: $M_n = M_{B,n-1} + M_{A,n}$.

Since each beam element is to be in equilibrium, we can apply equations 3.4.3-5 to elements n-1 and n in fig 3.4.4, and get

$$Q_n = -\frac{K_{n-1}}{\ell_{n-1}}\theta_{n-1} - 6\left\{\frac{K_{n-1}}{\ell_{n-1}} - \frac{K_n}{\ell_n}\right\}\theta_n + 6\frac{K_n}{\ell_n}\theta_{n+1} - \frac{12K_{n-1}}{\ell_{n-1}^2}\delta_{n-1} + \ldots$$

$$\ldots + 12\left\{\frac{K_{n-1}}{\ell_{n-1}^2} + \frac{K_n}{\ell_n^2}\right\}\delta_n - 12\frac{K_n}{\ell_n^2}\delta_{n+1} \qquad (3.4.6)$$

and

$$M_n = 2K_{n-1}\theta_{n-1} + 4\{K_{n-1} + K_n\}\theta_n + 2K_n\theta_{n+1} + 6\frac{K_{n-1}}{\ell_{n-1}}\delta_{n-1} - \ldots$$

$$\ldots - 6\left\{\frac{K_{n-1}}{\ell_{n-1}} - \frac{K_n}{\ell_n}\right\}\delta_n - 6\frac{K_n}{\ell_n}\delta_{n+1} \qquad (3.4.7)$$

At each node, some of these quantities will be known and some unknown, depending whether the node is at a bay end or mid-point. Suppose that node n is a bay mid-point; then the applied force Q_n is prescribed and there is no applied moment; the displacement and angle of rotation δ_n and θ_n are unknown. At the bay end-point nodes n-1 and n+1, which are attached to footings, the footing reactions R_{n-1} and R_{n+1} are the unknown forces, and the

angles of rotation θ_{n-1} and θ_{n+1} are also unknown. As we assume the footings to be pin-jointed to the beam, there is no moment created. Thus, at each node we have precisely two knowns and two unknowns. For the four nodes n-2, ... , n+1, as shown in fig 3.4.5, we may list the variables as follows:

<u>known</u>: Q_{n-2}, M_{n-2}, δ_{n-1}, M_{n-1}, Q_n, M_n, δ_{n+1}, M_{n+1} .

<u>unknown</u>: δ_{n-2}, θ_{n-2}, R_{n-1}, θ_{n-1}, δ_n, θ_n, R_{n+1}, θ_{n+1}.

As node n is the mid-point of the bay, we know that $K_{n-1} = K_n$ and $\ell_{n-1} = \ell_n$. Write these quantities as K_m and ℓ_m, on the assumption that we are dealing with the m'th bay (so that $\ell_{n-2} \equiv \ell_{m-1}$, etc).

Then applying equations 3.4.6-7 to the two nodes n-1 and n, and rearranging with the unknowns on the left-hand-side, we obtain the matrix equation

$$\underset{\sim}{T}_m \underset{\sim}{u}_m = \underset{\sim}{L}_m \underset{\sim}{v}_m \qquad (3.4.8)$$

where

$$\underset{\sim}{u}_m = (\delta_{n-2}\ \theta_{n-2}\ R_{n-1}\ \theta_{n-1}\ \delta_n\ \theta_n\ R_{n+1}\ \theta_{n+1})^T$$

$$\underset{\sim}{v}_m = (Q_{n-2}\ M_{n-2}\ \delta_{n-1}\ M_{n-1}\ Q_n\ M_n\ \delta_{n+1}\ M_{n+1})^T$$

and T_m and L_m are 8×8 matrices as given in fig 3.4.6. At the ends of the beam, ie for $m = 1$ and $m = M$, we obtain the matrices T_1, L_1, T_M, L_M as given in fig 3.4.7, with

$$\underset{\sim}{u}_1 = (R_1\ \theta_1\ \delta_2\ \theta_2\ R_3\ \theta_3)^T$$

$$\underset{\sim}{v}_1 = (\delta_1\ M_1\ Q_2\ M_2\ \delta_3\ M_3)^T$$

$$\underset{\sim}{u}_M = (\delta_{N-1}\ \theta_{N-1}\ R_N\ \theta_N)^T$$

$$\underset{\sim}{v}_M = (Q_{N-1}\ M_{N-1}\ \delta_N\ M_N)^T \qquad (3.4.9)$$

Node N is at the end of the beam, so $N = 2M - 1$.

These equations are assembled to give the global matrix equation for the equilibrium of the whole beam, viz

$$\underset{\sim}{T}\underset{\sim}{u} = \underset{\sim}{L}\underset{\sim}{v} \qquad (3.4.10)$$

where

$$\underset{\sim}{u} = (R_1\ \theta_1\ \delta_2\ \theta_2\ R_3\ \theta_3\ \ldots\ldots\ \delta_{N-1}\ \theta_{N-1}\ R_N\ \theta_N)^T$$

$$\underset{\sim}{v} = (\delta_1\ M_1\ Q_2\ M_2\ \delta_3\ M_3\ \ldots\ldots\ Q_{N-1}\ M_{N-1}\ \delta_N\ M_N)^T$$

and the matrices T, L are assembled from the submatrices $\{T_m\}$, $\{L_m\}$ as indicated in fig 3.4.8. This matrix equation has order $4M - 2$ where M is the number of footings. Unless the bay lengths and stiffnesses vary considerably along the beam, there has been found to be no singularity problems in inverting T by Gaussian elimination and using the flexibility matrix $T^{-1}L$ to premultiply the vector of known variables $\underset{\sim}{v}$ at each timestep and obtain the unknowns in $\underset{\sim}{u}$. The applied moments are always zero in the situation under consideration, although the analysis allows for non-zero moments also.

Loads applied directly above the footings may be included right at the end, by adding them to the footing reactions found from equation 3.4.10 (care being taken to first reverse the sign of the reactions, since these act upwards on the beam but downwards on the footing).

Subroutine STIFF to form the flexibility matrix $T^{-1}L$ initially, and STRUCT to perform the matrix multiplication at each timestep, have been written and included in the program FDTIM.

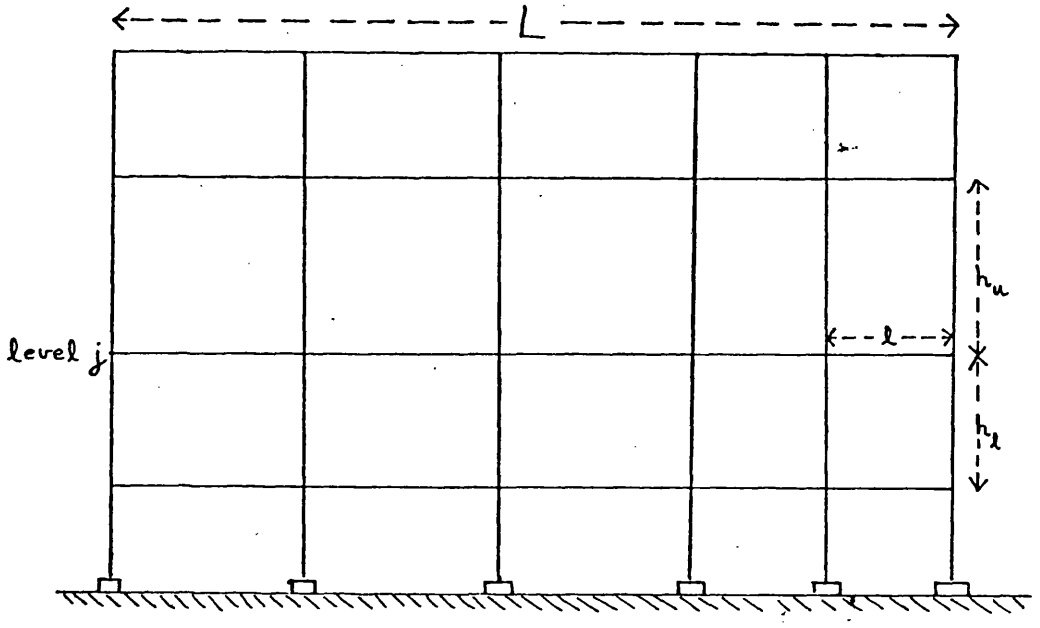


Fig. 3.4.1 Framed structure on footings

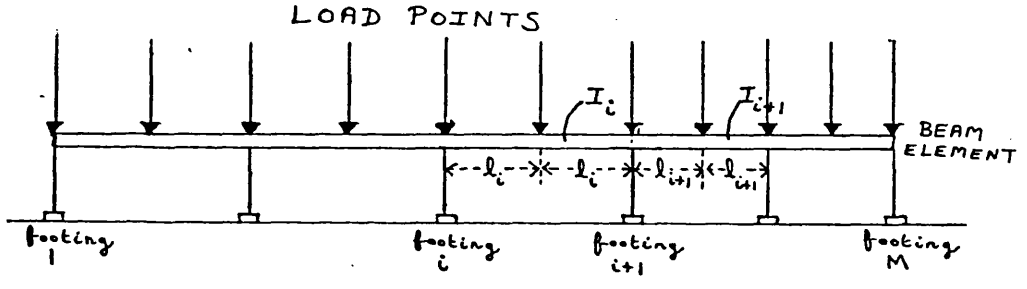


Fig. 3.4.2 Single-storey multi-bay portal

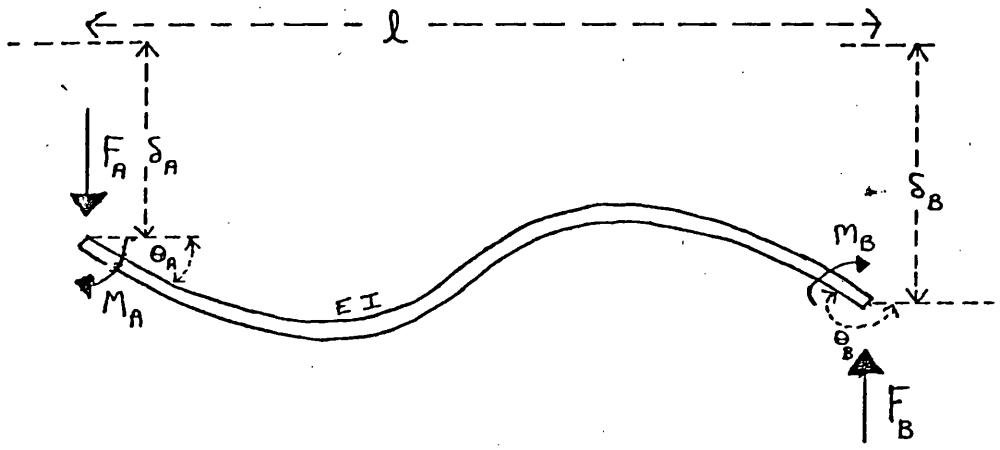


Fig. 3.4.3 Elastic bar, with forces and moments

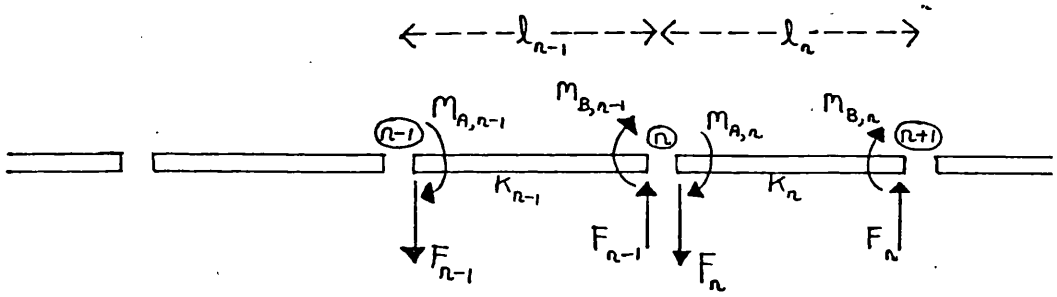


Fig. 3.4.4 Section of beam divided into elements

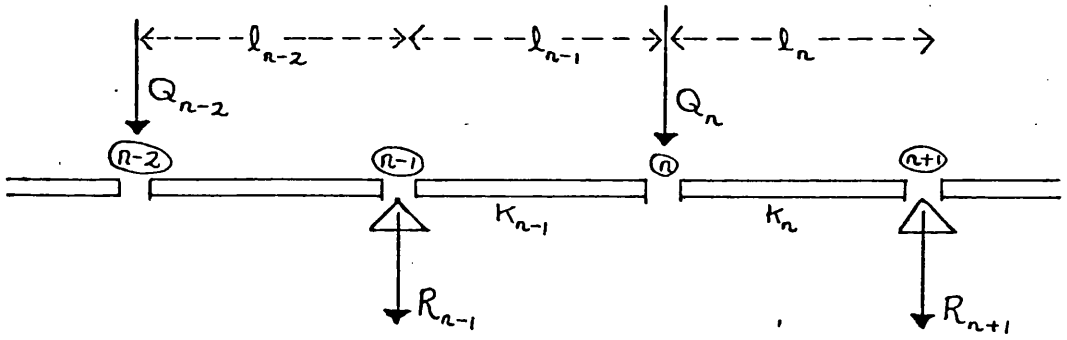


Fig. 3.4.5 Typical bay with loads Q and reactions R

$$T_m = \begin{pmatrix} 12\frac{K_{m-1}}{\ell_{m-1}^2} & 6\frac{K_{m-1}}{\ell_{m-1}} & 1 & 6\left[\frac{K_{m-1}}{\ell_{m-1}} - \frac{K_m}{\ell_m}\right] & 12\frac{K_m}{\ell_m^2} & -6\frac{K_m}{\ell_m} & 0 & 0 \\ -6\frac{K_{m-1}}{\ell_{m-1}} & -2K_{m-1} & 0 & -4[K_{m-1}+K_m] & 6\frac{K_m}{\ell_m} & -2K_m & 0 & 0 \\ 0 & 0 & 0 & 6\frac{K_m}{\ell_m} & -24\frac{K_m}{\ell_m^2} & 0 & 0 & -6\frac{K_m}{\ell_m} \\ 0 & 0 & 0 & -2K_m & 0 & -8K_m & & -2 \end{pmatrix}$$

$$L_m = \begin{pmatrix} 0 & 0 & 12\left[\frac{K_{m-1}}{\ell_{m-1}^2} + \frac{K_m}{\ell_m^2}\right] & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -6\left[\frac{K_{m-1}}{\ell_{m-1}} - \frac{K_m}{\ell_m}\right] & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -12\frac{K_m}{\ell_m^2} & 0 & -1 & 0 & -12\frac{K_m}{\ell_m^2} & 0 \\ 0 & 0 & 6\frac{K_m}{\ell_m} & 0 & 0 & -1 & -6\frac{K_m}{\ell_m} & 0 \end{pmatrix}$$

Fig. 3.4.6 Matrices in equation 3.4.8

$$T_1 = \begin{pmatrix} 1 & -\frac{6K_1}{\ell_1} & \frac{12K_1}{\ell_1^2} & -\frac{6K_1}{\ell_1} & 0 & 0 \\ 0 & -4K_1 & \frac{6K_1}{\ell_1} & -2K_1 & 0 & 0 \\ 0 & \frac{6K_1}{\ell_1} & -\frac{24K_1}{\ell_1^2} & 0 & 1 & -\frac{6K_1}{\ell_1} \\ 0 & -2K_1 & 0 & -8K_1 & 0 & -2K_1 \end{pmatrix}$$

$$L_1 = \begin{pmatrix} \frac{12K_1}{\ell_1^2} & 0 & 0 & 0 & 0 & 0 \\ \frac{6K_1}{\ell_1} & -1 & 0 & 0 & 0 & 0 \\ -\frac{12K_1}{\ell_1^2} & 0 & -1 & 0 & -\frac{12K_1}{\ell_1^2} & 0 \\ \frac{6K_1}{\ell_1} & 0 & 0 & -1 & -\frac{6K_1}{\ell_1} & 0 \end{pmatrix}$$

$$T_M = \begin{pmatrix} \frac{12K_{M-1}}{\ell_{M-1}^2} & \frac{6K_{M-1}}{\ell_{M-1}} & 1 & \frac{6K_{M-1}}{\ell_{M-1}} \\ -\frac{6K_{M-1}}{\ell_{M-1}} & -2K_{M-1} & 0 & -4K_{M-1} \end{pmatrix}$$

$$L_M = \begin{pmatrix} 0 & 0 & \frac{12K_{M-1}}{\ell_{M-1}^2} & 0 \\ 0 & 0 & -\frac{6K_{M-1}}{\ell_{M-1}} & -1 \end{pmatrix}$$

Fig. 3.4.7 Matrices in equation 3.4.8, for beam ends

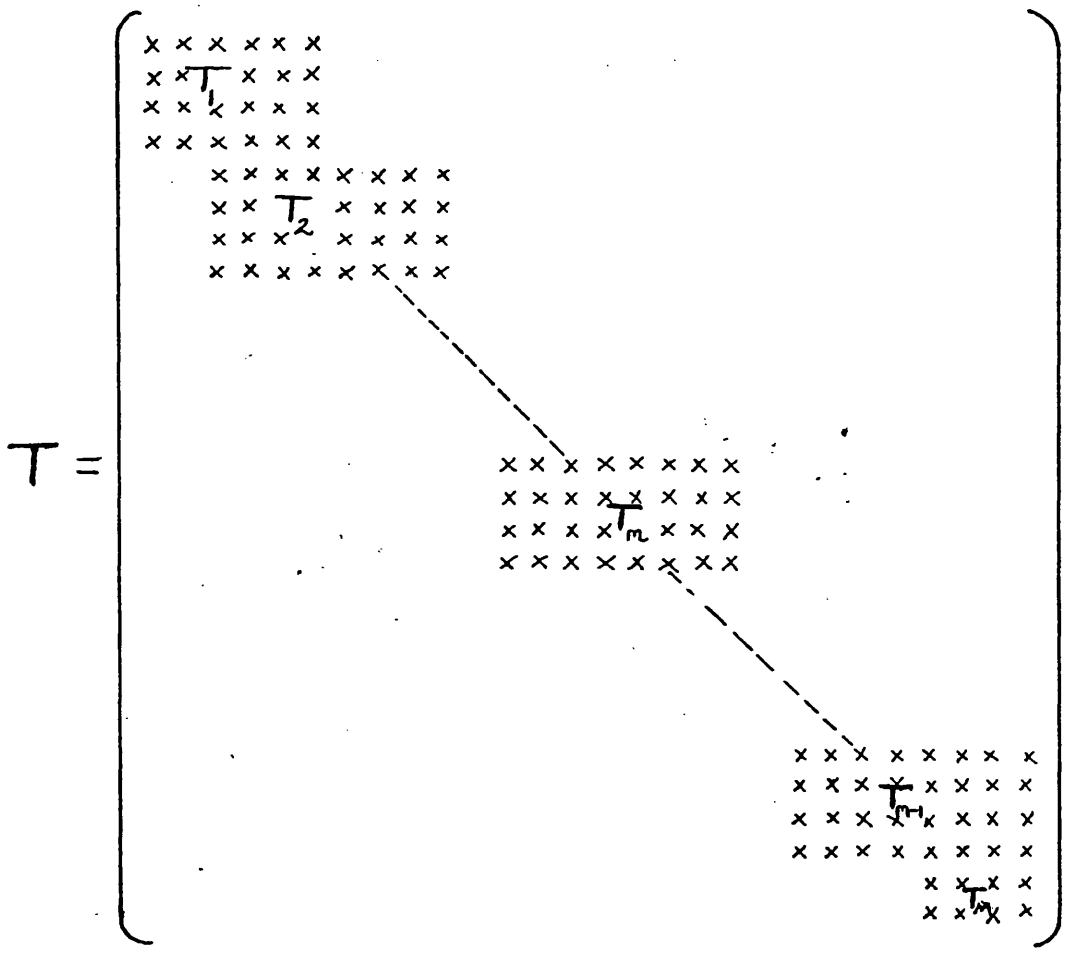


Fig. 3.4.8 Assembly into global matrix

3.5 Conclusion : the program FDTIM

We conclude by indicating the way in which the theory of chapters 2 and 3 is used in the program FDTIM. A detailed program specification is given in Appendix A. A simplified flow-diagram of the program is shown in fig 3.5.1.

Following input of the soil and structural data in the subroutine INPUT, subroutine STRUCT is called to evaluate the structural flexibility matrix $T^{-1}L$ of §3.4. (For independently loaded footings, this is omitted.)

The prescribed loads are then read, either directly as footing loads in the case of independent footings, or for linked footings as beam loads which are converted to footing loads using the structural flexibility matrix by subroutine STRUCT. The loads are converted to average footing pressures, and subroutine BOUSS or FILON is called, for each footing, to find the distribution of total stress increments, using the theory of §2.2. Four types of footing loads are allowed for, as shown in fig 3.5.2; by a combination of uniform and wedge-shaped loads, structures such as motorway embankments may also be modelled.

Following this a total stress analysis is carried out, with the undrained Young's modulus found from equation 2.4.7, to find the immediate vertical strains at each mesh-point. Since $\nu_u = 0.5$, these are given by

$$\varepsilon_{z_{ij}} = \frac{3}{4E_u}\{ \Delta\sigma_{z_{ij}} - \Delta\sigma_{x_{ij}} \} \qquad (3.5.1)$$

The strains are integrated to obtain the displacements at each mesh-point, using equation 2.3.11.

Thus the footing displacements are found. For the case of linked footings, these displacements will cause a redistribution of the load, and it is necessary to loop back to the statement calling subroutine STRUCT, to evaluate the new footing loads, and to perform the whole stress-displacement analysis again. The iteration continues until

$$\max_j \left| w_{j,n} - w_{j,n+1} \right| < e \qquad (3.5.2)$$

where $w_{j,n}$ is the displacement at footing j at the n'th iteration, and e is a specified small number. In practice, it is seldom necessary to perform more than three or four iterations. Iteration is of course unnecessary for independent footings. With the old and new stresses at each mesh-point now known, the induced pore-pressure increments can be found using equations 2.4.12-13.

Timestepping can now begin. Excess pore-pressures at permeable boundaries are set to zero before calling subroutine ADIDE, which sets up and solves the tridiagonal matrix equations 3.3.17-18 with the appropriate boundary conditions. An effective stress analysis now gives strains and displacements. Since the total stresses remain constant, the elasticity equation 2.3.10 reduces to

$$\varepsilon_{z_{ij}} = \frac{-(1+\nu)(1-2\nu)}{E} \Delta U_{ij} \qquad (3.5.3)$$

using the drained parameters E, ν. Displacements are obtained by numerical integration as before, and added to the previous displacements. There is now a return to the test 3.5.2, and if sufficient change in the footing displacements has occurred over the timestep, iteration over the undrained analysis must occur again. It has been found necessary to perform at least one iteration after each timestep, otherwise footing displacement changes negligible in themselves can build up over a number of timesteps without the necessary load redistribution process occurring.

At any time during the consolidation process, new loads may be added, so that a sequence of construction can be modelled with more accuracy than in the usual method using drained analyses, which assumes all excess pore-pressures to have dissipated before each new construction stage.

Since the diffusion process is one which decays with time, it is useful to increase the size of the timestep as the process continues and the changes become smaller and slower. In FDTIM, the timestep is multiplied by a specified factor after a set number of steps, reverting to the initial timestep when a

further load is applied.

Numerical results from the use of FDTIM are given in Chapter 6, where the performance of the program is compared with that of the finite element program FINTIM, which will now be described.

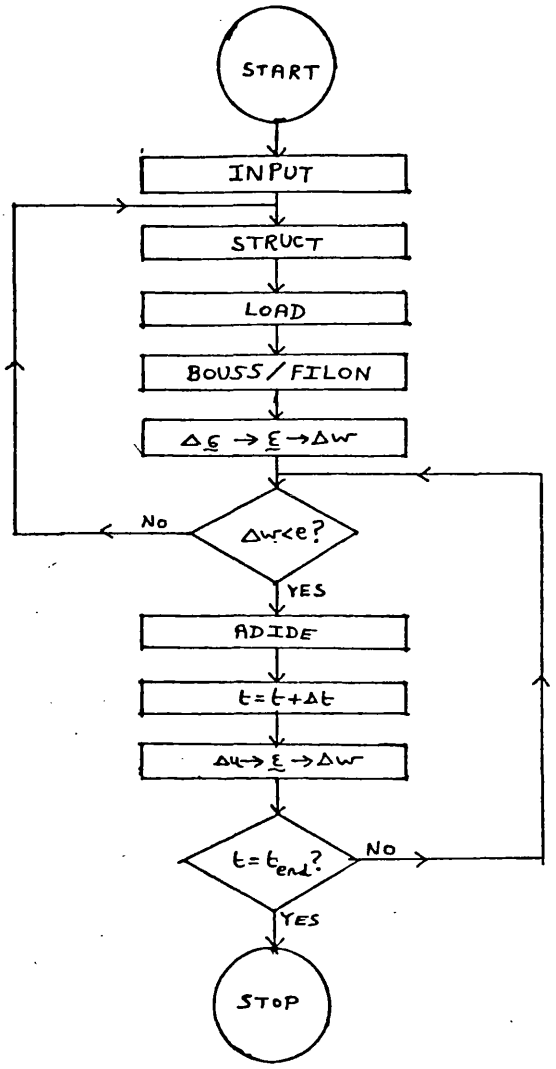


Fig. 3.5.1 Simplified flow-diagram: FDTIM

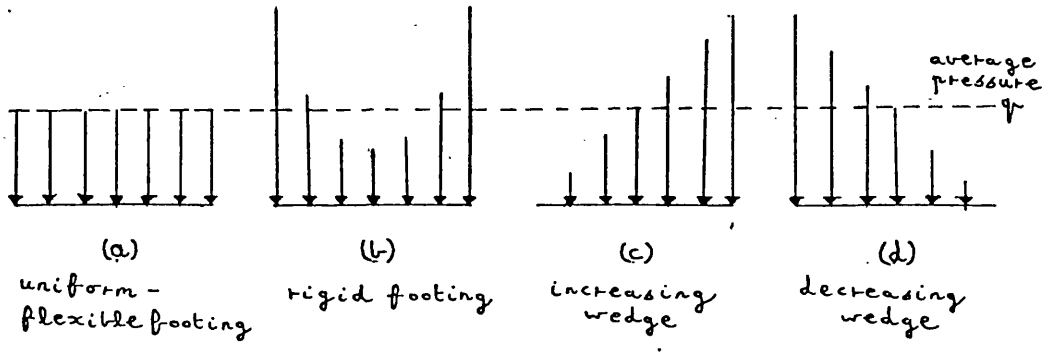


Fig. 3.5.2 Footing load types

CHAPTER 4. - THE FINITE ELEMENT MODEL

4.1 Introduction

In this chapter we present a second approach to the soil structure interaction problem, namely by modelling both the soil and the structure by means of finite elements. In this way, the behaviour of the soil and of the structure are coupled together automatically, and there is no need for the approximations and iteration developed for the program FDTIM in the previous two chapters. In addition there are the well-known advantages of generality of shape and boundary conditions which have made the finite element method such a powerful tool over a wide range of engineering problems.

There is copious literature on the finite element method, from practical handbooks such as Fenner (1975) to texts on the mathematical theoretic aspects such as Mitchell and Wait (1977), as well as wide-ranging reviews like Zienkiewicz (1977), and it is not proposed to give more than a brief outline here.

A finite element model of the problem to be analysed is first constructed by discretizing it into a series of elements connected at a finite number of points known as nodes. Then the equilibrium equations governing the material are applied to each element in turn to obtain an element stiffness matrix relating the system variables. It is assumed that these variables may be approximated at any point in the element by means of interpolation using the variable values at the node-points, and a set of shape-functions $\{N_i\}$. For example, consider a four-noded quadrilateral element as shown in Fig.4.2.1, with nodes at (±1, ±1) in the s - t local coordinate system. Then any variable v which ranges continuously over the element is approximated by a bilinear function:

$$v(s, t) = \sum_{i=1}^{4} v_i N_i \ (s,t) \qquad (4.1.1)$$

where v_i is the variable value at node i, and the shape function N_i equals 1 at node i and 0 at all other nodes. For this element, the bilinear shape functions are given by:

$$N_1 (s, t) = 1/4 (1 - s) (1 + t)$$

$$N_2 (s, t) = 1/4 (1 - s) (1 - t)$$

$$N_3 (s, t) = 1/4 (1 + s) (1 - t) \qquad (4.1.2)$$

$$N_4 (s, t) = 1/4 (1 + s) (1 + t)$$

In this way the element nodal values of the known and the unknown variables are related in a linear matrix equation involving the element stiffness matrix. In mechanics problems, the known variables are the external loads applied to the system. The unknown variables are generally taken as the displacements of the system - this is known as the displacement (or stiffness) method, and will be used here. The stress (or equilibrium) method takes stesses as the unknowns, and the mixed method takes both stresses and displacements as unknowns. Comparisons between the different methods may be found in Desai and Abel (1972). The element stiffness equation will thus be of the form:

$$K_e \underset{\sim}{\delta}_e = \underset{\sim}{f}_e \qquad (4.1.3)$$

where K_e is the element stiffness matrix,

$\underset{\sim}{\delta}_e$ is the vector of unknown nodal displacements,

and $\underset{\sim}{f}_e$ is the vector of known external nodal forces.

The element stiffness matrices and force vectors are assembled into a global stiffness equation, which is then solved to yield the unknown nodal displacements.

In § 4.2 the details of this process applied to the plane strain linear elastic soil problem are given, and its implementation in the computer program FINEPAK is described. The adaptation of this work to model the consolidation of a soil with time is dealt with in § 4.3, and the method of including structural elements in these models is described in § 4.4.

4.2 Drained Analysis by Finite Elements: Program FINEPAK

Theory

Perhaps the clearest outlines of the finite element method applied to a linear elastic continuum are in the introductory chapters of Desai and Christian (1977) and of Hinton and Owen (1977). The latter work describes in detail a computer program for soil mechanics applications upon which the program FINEPAK is based, and similar notation is used in this thesis. We will outline the derivation of the stiffness matrix for the two dimensional plane strain situation.

We may denote the independent stresses at any point in the continuum by a stress vector:

$$\underset{\sim}{\sigma} = (\sigma_x \ \sigma_z \ \tau_{xz})^T$$

and the corresponding strains by the strain vector:

$$\underset{\sim}{\varepsilon} = (\varepsilon_x \ \varepsilon_z \ \gamma_{xz})^T$$

The displacements in the x and z directions respectively are u and w, held in the displacement vector:

$$\underset{\sim}{\delta} = (u \ w)^T$$

If these stresses and strains are caused by body forces $\underset{\sim}{b} = (b_x \ b_z)^T$ and surface tractions $\underset{\sim}{p} = (p_x \ p_z)^T$ applied to the continuum, the total potential energy of the system π is given by:

$$\pi = \tfrac{1}{2} \iiint_V \underset{\sim}{\sigma}^T \underset{\sim}{\varepsilon} \, dV - \iiint_V \underset{\sim}{\delta}^T \underset{\sim}{b} \, dV - \int_S \underset{\sim}{\delta}^T \underset{\sim}{p} dS \qquad (4.2.1)$$

where V denotes the volume and S theloaded surface area. Using the shape functions $\{N_i\}$ as described in § 4.1, we may write the displacement $\underset{\sim}{\delta}$ at any point in terms of the nodal displacement vector $\underset{\sim}{\delta}^e$:

$$\underset{\sim}{\delta} = N\underset{\sim}{\delta}^e \qquad (4.2.2)$$

Strains are defined in terms of the partial derivatives of displacements, which enables us to form a strain matrix B such that:

$$\underset{\sim}{\varepsilon} = B\,\underset{\sim}{\delta}^{e} \qquad (4.2.3)$$

The forms of these matrices for the case of a four-noded quadrilateral element are given in Fig.4.2.1.

The elasticity equations 2.3.1 - 3 give us the elasticity matrix D relating stresses to strains:

$$\underset{\sim}{\sigma} = D\,\underset{\sim}{\varepsilon} \qquad (4.2.3)$$

In isotropic plane strain,

$$D = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \qquad (4.2.5)$$

Substituting into equation 4.2.1 to find the potential energy in an element e:

$$\pi_e = \tfrac{1}{2} \iiint_{V_e} \underset{\sim}{\delta}^{e^T} B^T D B\, \underset{\sim}{\delta}^{e}\, dV - \iiint_{V_e} \underset{\sim}{\delta}^{e^T} N^T \underset{\sim}{b}\, dV$$

$$- \int_{S_e} \underset{\sim}{\delta}^{e^T} N^T \underset{\sim}{p}\, dS \qquad (4.2.6)$$

The potential energy function provides us with the required functional, which is to be minimised over the whole system with respect to the nodal displacements for equilibrium. Now:

$$\frac{\partial \pi_e}{\partial \underset{\sim}{\delta}^{e}} = \iiint_{V_e} (B^T D B)\, \underset{\sim}{\delta}^{e}\, dV - \iiint_{V_e} N^T \underset{\sim}{b}\, dV - \int_{S_e} N^T \underset{\sim}{p}\, dS \qquad (4.2.7)$$

and the total potential energy of the system is simply the sum of the element energies, so by setting $\frac{\partial \pi}{\partial \underset{\sim}{\delta}} = 0$ we obtain:

$$K\,\underset{\sim}{\delta} = \underset{\sim}{f} \qquad (4.2.8)$$

where K is the global stiffness matrix assembled from the element stiffness matrices $K_e = \iiint_{V_e} B^T D B\, dV$, $\underset{\sim}{\delta}$ is the vector of nodal displacements for the whole system , and $\underset{\sim}{f}$ is the equivalent nodal force vector assembled from the element force vectors

$$\underset{\sim}{fe} = \iiint_{V_e} N^T \underset{\sim}{b}\, dV + \int_{S_e} N^T \underset{\sim}{p}\, dS.$$

The equilibrium equation 4.2.8 may alternatively be derived through application of the Principle of Virtual Work (see for

example Hinton and Owen (1977), p.16). Derivation of the specific forms of the D, B and N matrices for different types of element and situations other than plane strain, is straightforward.

Program FINEPAK

The author received in February 1978 a copy of the finite element program FINEPAK from Dr. D. J. Naylor of the Department of Civil Engineering, University College of Swansea. This program had been developed by Dr. Naylor based on the theory and notation contained in Hinton and Owen (1977), and the reader is referred to this text for a full explanation of the program structure. Practical notes and operating instructions are contained in the handbook by Naylor (1977); a brief outline will be given here.

The program consists of a short master program, to be written by the user, calling five main subroutines, which in turn make use of ten specialised 'satellite' subroutines for tasks such as, for example, constructing the D and B matrices, or performing Gaussian integration.

The program may be used for plane stress, plane strain, axisymmetric or three-dimensional problems, and allows a total of nine different types of element to be used, as shown in Fig.4.2.2. Elements of different types may be joined together.

The five main subroutines are:

1. INPUT. This reads, and checks for consistency, all data except applied forces.

2. STIFF. This evaluates for each element the stiffness matrix K_e defined in equation 4.2.8. The elasticity and strain matrices D and B are found, and their product $B^T DB$ formed, for the Gauss points of the element. The weighted sum of the Gauss point values forms the element stiffness matrix ESTIF which is written to file. (One-point or two-point Gaussian integration may be used; all work in this thesis has used the two-point integration rule).

3. LOAD. This reads in the applied forces, which may be normal or tangential surface loads, body forces, nodal loads

or specified displacements. Equivalent nodal forces are found, and assembled in the global force vector ASLOD, while specified displacements are assembled in SPDIS.

4. FRONT. This solves the global stiffness matrix equation 4.2.8. The frontal method of solution, as originated by Irons (1970), is used. Essentially, this is a form of Gaussian elimination for large symmetric systems, where the stiffness matrix has to be assembled from a number of smaller submatrices. It is thus particularly suited to the finite element method. The important point is that the variables are eliminated at the same time as the equations are assembled, so that it is not necessary to hold the global stiffness matrix in store. An excellent description of the method is given in Chapter 8 of Hinton and Owen (1977). A characteristic of FINEPAK which becomes of great importance in the time-dependent form is its capacity to deal with variable degrees of freedom per node. There is also a facility for re-solutions of the same stiffness matrix with different right hand sides, without having to re-assemble the stiffnesses.

5. OUTPUT. Nodal forces (including reactions at nodes with specified displacements) and displacements are output. Stresses and strains at element Gauss points may also be output if required (this is useful for non-linear stress-strain laws, see Chapter 5).

Implementation

FINEPAK, which was written for use on the ICL 1904S computer at Swansea, was implemented by the author to run on the ICL System 4 and 2980 machines of the South West Universitites Computer Network, using Double Precision arithmetic. Identical output for a set of test data supplied by Dr Naylor was obtained.

The master program was adapted from that provided with the FINEPAK program, to allow a number of loads to be applied successively. Each load can be applied in a specified number of equal increments.

A facility for performing analyses of undrained, immediate settlements was added. The theory for this is given in Naylor (1974):

all that is necessary is to replace the D matrix of equation 4.2.5 by D_u, given by:

$$D_u = D + K_a \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (4.2.9)$$

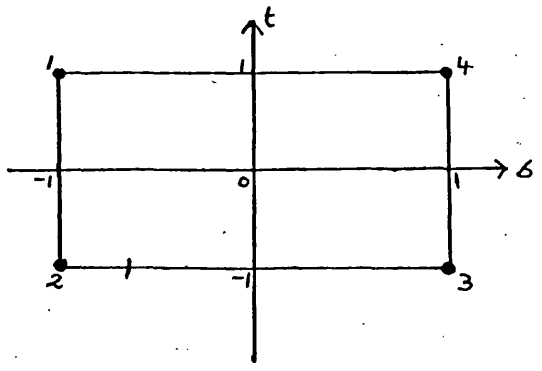
K_a is the apparent bulk modulus of compressibility of the pore fluid, which is added to the direct stress components of D, and is given by:

$$\frac{1}{K_a} = \frac{n}{K_f} + \frac{1 - n}{K_s} \qquad (4.2.10)$$

where K_f is the pore water bulk modulus

K_s is the bulk modulus of the soil skeleton

and n is the porosity of the soil.



Shape functions $\{N_i(s,t) \mid i=1,4\}$

$$\underset{\sim}{\delta}^e = (u_1 \;\; w_1 \;\; u_2 \;\; w_2 \;\; u_3 \;\; w_3 \;\; u_4 \;\; w_4)^T$$

$$N = \begin{pmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{pmatrix}$$

$$B = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_2}{\partial z} & 0 & \frac{\partial N_3}{\partial z} & 0 & \frac{\partial N_4}{\partial z} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial z} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial z} & \frac{\partial N_4}{\partial x} \end{pmatrix}$$

Fig. 4.2.1 Element matrices

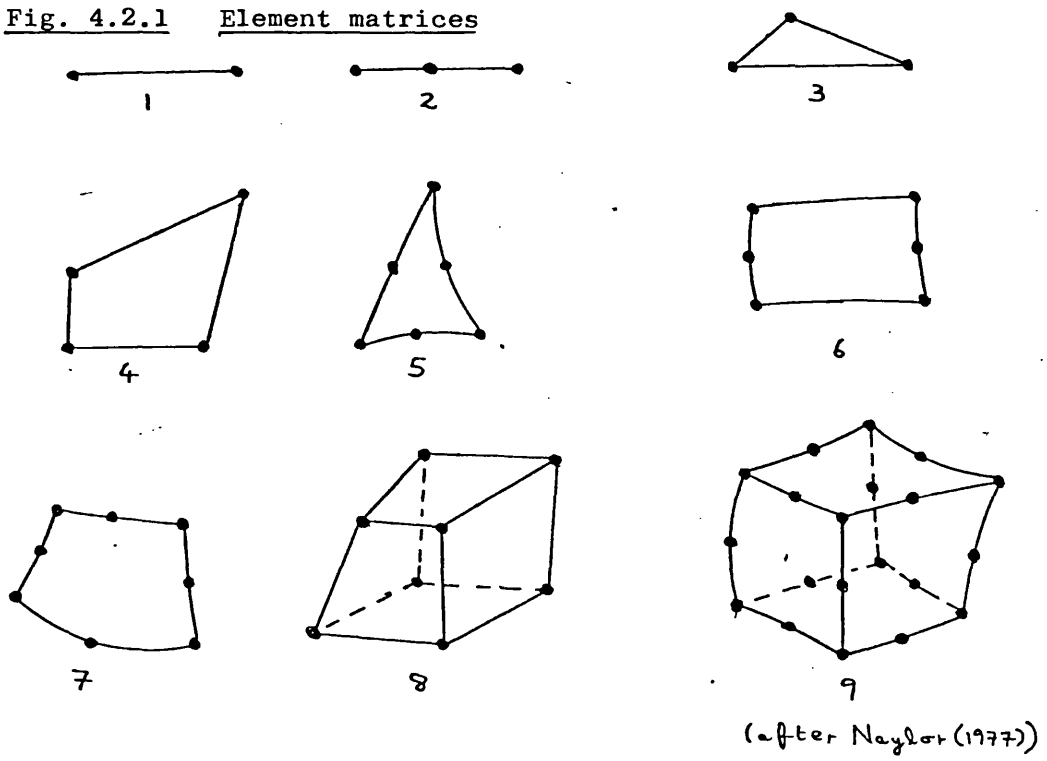


Fig. 4.2.2 FINEPAK element types

4.3 Modelling consolidation by finite elements: program FINETIM

To determine the displacements at any time when excess pore-pressures p are present (we shall follow for the remainder of this chapter the general practice of using p instead of u to denote pore-pressure in finite element theory. This is to avoid confusion with displacement in the x-direction), we have to perform an effective stress analysis using the drained soil parameters E and ν. The effective stress vector $\underset{\sim}{\sigma}' = (\sigma_x' \ \sigma_z' \ \tau_{xz}')^T$ at any point with pore-pressure p is related to the total stress vector σ by:

$$\underset{\sim}{\sigma}' = \underset{\sim}{\sigma} - (p \ p \ o)^T \qquad (4.3.1)$$

The pore-pressure p may be expressed in terms of the nodal pore-pressure vector $\underset{\sim}{p}^e$ by use of the shape functions $\{N_i \mid i = 1,..,n\}$:

$$\begin{pmatrix} p \\ p \\ o \end{pmatrix} = MN\underset{\sim}{p}^e \qquad (4.3.2)$$

where $M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$ and $N = \begin{pmatrix} N_1 & N_2 & \ldots\ldots & N_n \\ N_1 & N_2 & \ldots\ldots & N_n \end{pmatrix}$

Replacing $\underset{\sim}{\sigma}$ by $\underset{\sim}{\sigma}'$ in equation 4.2.1 and following through the theory of § 4.2, using equations 4.3.1 and 4.3.2, we obtain:

$$K\underset{\sim}{\delta} - L^T\underset{\sim}{p} = \underset{\sim}{f} \qquad (4.3.3)$$

instead of equation 4.2.8. The stiffness matrix K is defined as in equation 4.2.8, and the matrix L is given by assembling the element matrices $L_e = \iiint_{V_e} N^T M^T B \, dV$.

If the pore- pressures are known, they may be transferred onto the right hand side and the matrix equation solved for the displacements. This suggests a method of solution of the consolidation problem by performing an undrained analysis as in § 4.2, deriving the corresponding undrained nodal excess pore-pressures by:

$$\underset{\sim}{p} = - K_f/n \ M^T B \ N \ \underset{\sim}{\delta}^e \qquad (4.3.4)$$

(see Desai and Christian (1977), p.133) and then calculating the nodal pore-pressures after a timestep Δt by a finite

difference approximation of the equation of flow, viz.

$$-\frac{\partial \varepsilon_v}{\partial t} = \frac{k_x}{\gamma_w}\frac{\partial^2 p}{\partial x^2} + \frac{k_z}{\gamma_w}\frac{\partial^2 p}{\partial z^2} \qquad (4.3.5)$$

By alternating between finite difference timestepping for the new pore-pressures, and finite element solutions for the new displacements, the process of consolidation may be modelled. This procedure was developed and used in plane strain problems by Christian and Boehmer (1970) and Christian et al (1972). However, a more elegant method proposed by Sandhu and Wilson (1969) involves deriving a second matrix equation from the equation of flow, and coupling this and the equilibrium equation 4.3.3 together, solving at each timestep for the displacements and pore-pressures as unknowns simultaneously. This method has since been used with success in a variety of problems, and is the procedure adopted in the program FINETIM.

By substituting equations 4.2.2 - 5 and 4.3.2 into the equation of flow, 4.3.5, and including a term to allow for the compressiblity of the pore-water, one obtains:

$$-H\,\underset{\sim}{p} - L\,\underset{\sim}{\delta}' - S\,\underset{\sim}{p}' = \underset{\sim}{r} \qquad (4.3.6)$$

Where L is as defined in equation 4.3.3, and the H and S matrices are assembled from element matrices H_e, S_e where:

$$H_{e_{ij}} = \iint_v \left(\frac{k_x}{\gamma_w}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{k_z}{\gamma_w}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z} \right) dV$$

$$S_e = \frac{n}{K_f} \iint_v N^T N \, dV$$

k_x, k_z are the soil element permeabilities,

n is the porosity,

K_f is the pore water bulk modulus,

$\underset{\sim}{r}$ has zero entries except on permeable boundaries (see below),

the dashed superscript ' denotes differentiation with respect to time.

Equations 4.3.3 and 4.3.6 may be combined to give:

$$\begin{pmatrix} K & -L^T \\ 0 & -H \end{pmatrix} \begin{pmatrix} \delta \\ \sim \\ p \\ \sim \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ -L & -S \end{pmatrix} \begin{pmatrix} \delta' \\ \sim \\ p' \\ \sim \end{pmatrix} = \begin{pmatrix} f \\ \sim \\ r \\ \sim \end{pmatrix} \qquad (4.3.7)$$

We shall denote the two composite matrices by X and Y respectively, the vector of unknowns by $\underset{\sim}{v}$ and the right hand side vector by $\underset{\sim}{a}$, so that equation 4.3.7 may be written;

$$X\underset{\sim}{v} + Y\underset{\sim}{v}' = \underset{\sim}{a} \qquad (4.3.8)$$

Time-discretization

In order to produce a numerical solution to equation 4.3.8, we must make a finite difference approximation for the derivatives. The simplest way is to use a forward difference approximation:

$$v' \approx [v_{n+1} - v_n] / \Delta t \qquad (4.3.9)$$

where v_n is the vector of unknowns at time t_n, and $\Delta t = t_{n+1} - t_n$, and then to evaluate the other vectors at some intermediate time:

$$\underset{\sim}{v} \approx \theta \underset{\sim}{v}_n + (1 - \theta)\underset{\sim}{v}_{n+1}$$

$$\underset{\sim}{a} \approx \theta \underset{\sim}{a}_n + (1 - \theta)\underset{\sim}{a}_{n+1} \qquad 0 \leqslant \theta \leqslant 1 \qquad (4.3.10)$$

Substituting into 4.3.8 and rearranging,

$$\left[(1 - \theta)X + \frac{1}{\Delta t} Y\right] \underset{\sim}{v}_{n+1} = \theta \underset{\sim}{a}_n + (1 - \theta)\underset{\sim}{a}_{n+1}$$

$$- \left[\theta X - \frac{1}{\Delta t} Y\right] \underset{\sim}{v}_n \qquad (4.3.11)$$

When a_n, a_{n+1} and v_n are known, this becomes a global matrix equation for the new unknowns v_{n+1}, with stiffness matrix:

$$(1 - \theta)X + \frac{1}{\Delta t} Y = \begin{pmatrix} (1 - \theta)K & -(1 - \theta)L^T \\ -\frac{1}{\Delta t} L & -(1 - \theta)H - \frac{1}{\Delta t} S \end{pmatrix} \qquad (4.3.12)$$

It is clear that this stiffness matrix is not symmetric, so that the frontal solution technique as implemented in subroutine FRONT of FINEPAK is inapplicable. Symmetry may be restored, however by first dividing equation 4.3.3 (the top row of the composite equation 4.3.7) by $(1 - \theta)$, and by multiplying equation 4.3.6 by Δt. The revised form of equation 4.3.11 then becomes:

$$\left[\bar{X} + Y\right] \underset{\sim}{v}_{n+1} = \begin{pmatrix} \bar{\theta} f_n + f_{n+1} \\ \theta \Delta t r_n + (1 - \theta)\Delta t r_n \end{pmatrix} - \left[\bar{\theta}\bar{X} - Y\right] v_n \qquad (4.3.13)$$

where:

$$\bar{X} = \begin{pmatrix} K & -L^T \\ 0 & -(1 - \theta)\Delta t\, H \end{pmatrix} \text{ and } \bar{\theta} = \frac{\theta}{1 - \theta}$$

and Y is unchanged. Now $\bar{X} + Y$ is a symmetric matrix.

Initial conditions

The immediate displacements and pore-pressures, resulting from the sudden application of a force vector $\underset{\sim}{q}$ at time t_o, is found by letting $\theta = \frac{1}{2}$ and $\Delta t \to 0$, with $\underset{\sim}{\delta}_o = \underset{\sim}{p}_o = \underset{\sim}{0}$:

$$\begin{pmatrix} K & -L^T \\ -L & -S \end{pmatrix} \begin{pmatrix} \underset{\sim}{\delta} \\ \underset{\sim}{p} \end{pmatrix} = \begin{pmatrix} \underset{\sim}{q} \\ \underset{\sim}{0} \end{pmatrix} \qquad (4.3.14)$$

References

It is hoped that the above exposition provides a clear, practical outline of consolidation theory by finite elements, comparable to that for drained analyses given in Hinton and Owen (1977) and followed in § 4.2. Derivations of the governing equations from variational principles may be found in Small et al (1975), and in Chapter 12 of Desai and Christian (1977) as well as in the original paper by Sandhu and Wilson (1969). The coupling of the equations and the time-discretization is discussed in lectures 1 and 8 of Naylor (1978), and in Chapter 3 of Desai and Christian (1977). The latter account, by Zienkiewicz, is the clearest to date, though suffering from a number of typographical errors. Cross-referencing between accounts is also hindered by the differences in sign convention used for stresses and for pore-pressures.

Choice of θ and Δt

With $0 < \theta \leqslant \frac{1}{2}$ the timestepping scheme is unconditionally stable for all values of Δt. For $\theta > \frac{1}{2}$, the scheme becomes conditionally stable, and Booker and Small (1975) have shown that if further the timestep is strictly increasing then the scheme is unstable for all Δt. The Crank-Nicholson scheme, with $\theta = \frac{1}{2}$, has proved satisfactory in practice, though Zienkiewicz (1977), Chapter 21, reports that this may become inaccurate for large Δt, and recommends the Galerkin scheme $\theta = 1/3$.

A different problem arises for very small values of Δt. Due to the discrepancy in order of magnitude between the elements of submatrix K (which involve the Young's Modulus E as a factor) and those of the submatrix $(1 - \theta)\Delta t\, H$ (involving the relative permeabilities as factors) in the composite stiffness matrix in

equation 4.3.13, there may be numerical problems due to loss of significant figures in the equation solution process. Note that the pore-water is very nearly incompressible, so the elements of matrix S will also be close to zero. Ghaboussi and Wilson (1973) have established that the minimum value of Δt to avoid this problem is:

$$\Delta t_{min} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \frac{\gamma_w}{k \times 10^n} \qquad (4.3.15)$$

for a uniform mesh, and an isotropic soil of permeability k, where n is the number of digits held in the computer.

This problem will be worsened in a soil-structure interaction problem such as the present one, where there will be structural elements with a Young's Modulus several orders of magnitude greater than that for the soil.

This source of error may be eliminated by use of a scaling factor c applied to the pore-pressures p. The iterative scheme we have developed (equation 4.3.13) is:

$$\begin{pmatrix} K & -L^T \\ -L & -(1-\theta)\Delta t.\, H \; -S \end{pmatrix} \begin{pmatrix} \underset{\sim}{\delta} \\ \underset{\sim}{p} \end{pmatrix}_{n+1} = \begin{pmatrix} \bar{\theta} \underset{\sim}{f}_n + \underset{\sim}{f}_{n+1} \\ \theta \Delta t \underset{\sim}{r}_n + (1-\theta)\Delta t \underset{\sim}{r}_{n+1} \end{pmatrix}$$

$$- \begin{pmatrix} \bar{\theta}K & -\bar{\theta}L^T \\ L & S - \theta.\Delta t.H \end{pmatrix} \begin{pmatrix} \underset{\sim}{\delta} \\ \underset{\sim}{p} \end{pmatrix} \qquad (4.3.16)$$

Let $\underset{\sim}{p}^* = \frac{1}{c} \underset{\sim}{p}$. Substituting into 4.3.16, and multiplying the lower set of equations by c to keep the stiffness matrix symmetric, we have:

$$\begin{pmatrix} K & -cL^T \\ -cL^T & -c^2 S - c^2(1-\theta)\Delta t H \end{pmatrix} \begin{pmatrix} \underset{\sim}{\delta} \\ \underset{\sim}{p}^* \end{pmatrix}_{n+1} = \begin{pmatrix} \bar{\theta} \underset{\sim}{f}_n + \underset{\sim}{f}_{n+1} \\ c\theta \Delta t \underset{\sim}{r}_n + c(1-\theta)\Delta t \underset{\sim}{r}_{n+1} \end{pmatrix}$$

$$- \begin{pmatrix} \bar{\theta}K & -c\bar{\theta}L^T \\ cL & c^2 S - c^2 \theta \Delta t H \end{pmatrix} \begin{pmatrix} \underset{\sim}{\delta} \\ \underset{\sim}{p}^* \end{pmatrix} \qquad (4.3.17)$$

The minimum value of Δt for the new stiffness matrix now becomes, by Ghaboussi and Wilson's analysis,

$$\Delta t^*_{min} = \frac{1}{c^2} \Delta t_{min} \qquad (4.3.18)$$

For a typical soil-structure interaction problem using kilograms/

metres/seconds as units, the elements of K will be of the order of 10^6 (for structural elements), and those of H will be of the order of 10^{-6} (for clay soil elements); a choice of $c \approx 10^6$ would therefore be appropriate, to raise all the elements of the stiffness matrix to the same order of magnitude. A facility for specifying c has been included in the program FINETIM, and the effect is seen by outputting in subroutine FRONT the sizes of the largest and smallest pivots occurring in the matrix equation solution process.

Choice of elements

In their original paper, Sandhu and Wilson (1969) proposed a composite triangular element, as shown in Fig.4.3.1(a). There are displacement degrees of freedom at all six nodes, but pore-pressure degrees of freedom at only the three corner nodes. The reason for this, is that with a quadratic variation of displacements over the element, the stresses and strains obtained by differentiating the displacements would vary linearly, and so the pore-pressure, which is also a stress, should vary linearly too. Ghaboussi and Wilson used a four-noded rectangular element, but introduced a 'fictitious' fifth node for the displacement interpolations, the degrees of freedom for which were later eliminated from the stiffness matrix, to satisfy the above argument. Later workers have used the composite quadrilateral isoparametric element in Fig.4.3.1(b), with eight nodes, of which four had pore-pressure degrees of freedom.

Yokoo et al (1971) experimented with the standard types of finite element, where all the unknowns had the same shape functions used on all the nodes. The effect of this is to produce an oscillatory distribution in space, of pore-pressures, when solving equation 4.3.14 for the initial undrained conditions. Mathematically, this appears to result from the ill-conditioned nature of the initial stiffness matrix in equation 4.3.14, where for an incompressible pore-fluid the submatrix S becomes the zero matrix. Once the timestepping process is started, however, the consolidation equation is solved satisfactorily, and the initial oscillations die away. Sandhu et al (1977) report that use of composite elements,

while avoiding this oscillatory problem, gives displacements which are somewhat less accurate during the early stages of consolidation (see lecture 8 of Naylor (1978)).

Use of composite elements provides an extra complication to the theory and programming, as two different sets of shape functions - one for the displacements and one for the pore-pressures - have to be used in assembling each element stiffness matrix. In the following section a method of smoothing the initial pore-pressure oscillations is proposed, which then enables the standard types of finite elements to be used and combined.

Smoothing of initial pore-pressures

A very similar oscillatory problem occurs in time-independent finite element analyses of materials with low compressibility (as is the case in undrained conditions), when the stresses at element nodes are required. In the displacement method, nodal stresses may be evaluated for each element once the displacements have been found, by use of equations 4.2.3 - 4. For nearly incompressible materials, it is found that different answers for the stresses at a node may be obtained from the different elements to which the node belongs. Further, the distribution of stresses over linear elements has a saw-tooth oscillation, and over quadratic elements a cupped oscillation. The phenomenon is discussed and illustrated in Naylor (1974), from which Fig.4.3.2(a) is taken. Hinton and Campbell (1974) suggest that for parabolic elements a bilinear smoothing process applied to each element can find the correct stresses; to eliminate nodal discontinuities, these element smoothings are assembled into a global smoothing process. Both these papers point out that the stresses at the Gauss points of the elements (found from the nodal stresses and shape functions) are in close agreement with the correct values. This suggests a smoothing process as illustrated in one dimension in Fig.4.3.2(b).For two-point Gaussian integration over the interval $[-1,1]$, the Gauss-points are located at $\pm 1/\sqrt{3}$ (the theory of Gaussian integration is described in, for example, Phillips and Taylor (1973), p.138 ff). The nodal values p_1, p_2, p_3 at $x = -1,0,1$ respectively may be smoothed to $\tilde{p}_1, \tilde{p}_2, \tilde{p}_3$ by

fitting a quadratic through the three points, then a straight line through the points on the curve at $x = \pm 1/\sqrt{3}$. One can show that:

$$\tilde{p}_1 = 2/3\ p_1 + 2/3\ p_2 - 1/3\ p_3$$

$$\tilde{p}_2 = 1/6\ p_1 + 2/3\ p_2 + 1/6\ p_3 \qquad (4.3.19)$$

$$\tilde{p}_3 = -\ 1/3\ p_1 + 2/3\ p_2 + 2/3\ p_3$$

For the two-dimensional case, the situation is more complicated. Consider a nine-noded quadrilateral element as shown in Fig.4.3.3. There are four Gauss-points, at $(\pm 1/\sqrt{3}, \pm 1/\sqrt{3})$, labelled I, II, III, IV. Let the unsmoothed nodal values be $p_1, \ldots, p_9$. Then we can fit a unique biquadratic through these points, viz.

$$p = 1/4x(x - 1)y(y + 1)p_1 - \tfrac{1}{2}x(x - 1)(y - 1)(y + 1)p_2 + 1/4x(x - 1)$$
$$y(y - 1)p_3 - \tfrac{1}{2}(x - 1)(x + 1)y(y - 1)p_4 + 1/4x(x + 1)y(y - 1)p_5$$
$$- \tfrac{1}{2}x(x + 1)(y + 1)(y - 1)p_6 + 1/4x(x + 1)y(y + 1)p_7 \qquad (4.3.20)$$
$$- \tfrac{1}{2}(x - 1)(x + 1)y(y + 1)p_8 + (x - 1)(x + 1)(y - 1)(y + 1)p_9$$

Evaluating this at the Gauss-points we obtain:

$$\underset{\sim}{g} = M_1 \underset{\sim}{p} \qquad (4.3.21)$$

where $\underset{\sim}{g} = (p_I\ p_{II}\ p_{III}\ p_{IV})^T$ is the vector of Gauss-point values, $\underset{\sim}{p} = (p_1\ p_2\ p_3\ \ldots\ p_9)^T$ is the vector of unsmoothed nodal values, and the matrix M_1 is shown in Fig.4.3.4.

Through $\underset{\sim}{g}$ we fit a unique bilinear surface of the form:

$$p = a_1 + a_2 x + a_3 y + a_4 xy. \qquad (4.3.22)$$

The coefficients $a_1, \ldots, a_4$ will be given by:

$$\underset{\sim}{g} = M_2 \underset{\sim}{a} \qquad (4.3.23)$$

where $\underset{\sim}{a} = (a_1\ a_2\ a_3\ a_4)^T$ and M_2 is as in Fig.4.3.4. Evaluating this surface at the nodes gives the smoothed values $\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_9$:

$$\underset{\sim}{\tilde{p}} = M_3\ \underset{\sim}{a} \qquad (4.3.24)$$

where $\underset{\sim}{\tilde{p}} = (\tilde{p}_1\ \tilde{p}_2\ \ldots\ \tilde{p}_9)^T$ and M_3 is given in Fig.4.3.4. Combining

equations 4.3.21, 4.3.23 and 4.3.24 we have the smoothing process:

$$\tilde{\underset{\sim}{p}} = M_3 M_2^{-1} M_1 \underset{\sim}{p} = M\underset{\sim}{p} \qquad (4.3.25)$$

where M is given in Fig.4.3.4

We have to eliminate the central node 9 to have a formula applicable to the eight-noded quadrilateral element. In order for the formula to reduce to the one-dimensional formula (equations 4.3.19) when the pore-pressure (which is always positive) is constant along one of the coordinate axes, we take:

$$p_9 = \text{Max}\left(\frac{p_2 + p_6}{2}, \frac{p_4 + p_8}{2}\right) \qquad (4.3.26)$$

For completeness, we may apply the principle, of a smoothing function of lesser degree than that of the element shape functions, to other quadrilateral elements. In the case of a four-noded quadrilateral, with bilinear shape functions, 2 x 2 Gaussian integration is exact, and we can simply average the nodal pore-pressures:

$$\tilde{p}_1 = \tilde{p}_2 = \tilde{p}_3 = \tilde{p}_4 = 1/4(p_1 + p_2 + p_3 + p_4) \qquad (4.3.27)$$

The intermediate case of a six noded quadrilateral,with midside nodes at nodes 2 and 5, we may treat as a one-dimensional case in the quadratic - integration direction, while averaging in the linear - interpolation, viz.

$$\tilde{p}_1 = \tilde{p}_6 = 1/3(p_1 + p_6) + 1/3(p_2 + p_5) - 1/6(p_3 + p_4)$$

$$\tilde{p}_2 = \tilde{p}_5 = 1/12(p_1 + p_6) + 1/3(p_2 + p_5) + 1/12(p_3 + p_4) \qquad (4.3.28)$$

$$\tilde{p}_3 = \tilde{p}_4 = -1/6(p_1 + p_6) + 1/3(p_2 + p_5) + 1/3(p_3 + p_4)$$

Applying the smoothing process to each element in turn, we obtain for any node N a series of smoothed pore-pressures $\tilde{p}_{N',e}$, one from each element e containing node N. For a unique smoothed value at node N, we average these, giving weightings according to the size of the elements:

$$\tilde{p}_N = \left(\sum_e \tilde{p}_{N,e}\Delta_e\right) \Big/ \sum_e \Delta_e \qquad (4.3.29)$$

where Δ_e is the area of element e.

The smoothing process has been programmed as a subroutine named SMOOTH which may be called after the initial displacements and pore-pressures have been found. It is important that once the pore-pressures have been smoothed, displacements compatible with these pore-pressures should be used in subsequent timestepping, otherwise instability arises. This is done by adding the smoothed pore-pressure values to the set of specified displacements and re-solving the stiffness equation for the initial conditions, to obtain new displacements.

Numerical examples of this process are given in Chapter 6, where it is shown that good agreement may be obtained with exact solutions.

Analogy with the method of collocation.

The 2 x 2 Gauss points exhibit similar optimal properties in the numerical solution of ordinary and partial differential equations by the method of collocation. In this method, we express the solution to the equation

$$L\,u(t) = f(t) \qquad (4.3.30)$$

in a given region, where L is a differential operator, as a linear combination of chosen basis functions, say:

$$u_n(t) = a_1\phi_1(t) + a_2\phi_2(t) + \ldots + a_n\phi_n(t) \qquad (4.3.31)$$

This solution is substituted into the governing differential equation 4.3.30 at a finite number of selected points $t_1, t_2, \ldots, t_n$ in the region, producing a set of linear equations:

$$a_1 L\phi(t_i) + a_2 L\phi_2(t_i) + \ldots + a_n L\phi_n(t_i) = f(t_i) \quad i = 1, \ldots, n \qquad (4.3.32)$$

which are solved for the coefficients $a_1, \ldots, a_n$.

The method is discussed in Prenter (1975), where it is shown (p.304) that for a second order boundary value problem, collocation at two-point Gaussian knots using a basis of piecewise cubic Hermite polynomials gives an algorithm with error of order h^4 (where h is the mesh size), compared with errors of order h^2 or better by collocation using cubic splines and their knots.

For partial differential equations the same improvement in accuracy results from using the 2 x 2 Gauss-points of a rectangular mesh, and expressing the function by piecewise bicubic Hermite polynomials - see Prenter and Russell (1976).

Such a procedure could be applied to the present problem, producing a smooth pore-pressure distribution with continuous first derivative everywhere. To obtain this continuity of slope at element boundaries, however, the method becomes implicit and involves forming, assembling and solving a global smoothing matrix equation. We only require the nodal values, and the explicit process described above has been found to produce sufficiently smooth pore-pressure values (see results in § 6.2) at a fraction of the cost.

Program FINETIM

Two features of the program FINEPAK were of particular importance in adapting it to a time-dependent form: one was the wide range of element types which could be used and joined together (see Fig.4.2.2), whereby in compiling each element stiffness matrix in subroutine STIFF a check is made on the element type and the appropriate set of shape functions $\{N_i\}$ used; the other is the facility for specifying the number of degrees of freedom independently at each node. Composite elements as in Fig.4.3.1 can be input by specifying two degrees of freedom at midside nodes, and three at corner nodes, the third degree of freedom being used for pore-pressures. In subrouting STIFF, having compiled the element load-displacement stiffness matrix K, a check is made on the element type as regards the pore-pressure degree of freedom, and if this is different from the displacement degrees of freedom element type (as in the case for composite elements) new shape functions and derivative functions are selected before compiling the H, L and S submatrices from the definitions given above. The element matrices X and Y as defined in equations 4.3.7 - 8 are assembled from the submatrices K, H, L and S.

It would be impractical to use the element stiffness matrices in the form shown so far, with all the nodal displacement degrees of freedom coming together, followed by the nodal pore-pressure

degrees of freedom; these matrices would have a very large bandwidth which would make the frontal solution very inefficient. We require the pore-pressure degree of freedom (if any) at node N to come immediately after the displacement degrees of freedom at that node, in the vector of unknowns. Thus, the columns of the X and Y matrices must be rearranged to effect this, and by also rearranging the rows in the same way the symmetry of the stiffness matrix when finally assembled can be retained. For each element, the X and Y matrices are written to a scratch file in subroutine STIFF.

The time-stepping scheme we shall use is given in equation 4.3.13; the element stiffness matrices are given by $\bar{X} + Y$. To convert X (read from the scratch file) to $\bar{X}$ in subroutine FRONT prior to solution, the elements of X corresponding to the submatrix H must be multiplied by $(1 - \theta)\Delta t$. The reason for doing this in FRONT, rather than immediately upon compiling X in STIFF, is that it is convenient to increase the size of Δt after every five steps or so in a typical consolidation problem, and in this way this can be done without the necessity of re-calling STIFF to form $\bar{X}$ again.

The right hand side of equation 4.3.13 can then be assembled, and the frontal solution proceeds unchanged.

Pore-pressure boundary conditions are very easy to handle. At permeable boundaries, the excess pore-pressure is zero, and is specified as a fixed degree of freedom in INPUT. There is a resulting non-zero reaction which appears in $\underset{\sim}{r}$. The natural boundary conditions of the finite element formulation correspond to an impermeable boundary, so that for nodes on impermeable boundaries no special consideration is necessary.

Provision is made for θ and the initial time-step to be specified, as well as the factor by which the time-step Δt may be increased after a set number of steps (see program specification in Appendix B). Several loads may be specified, togther with their times of application, and after each time-step the program checks if it is necessary to return to subroutine LOAD.

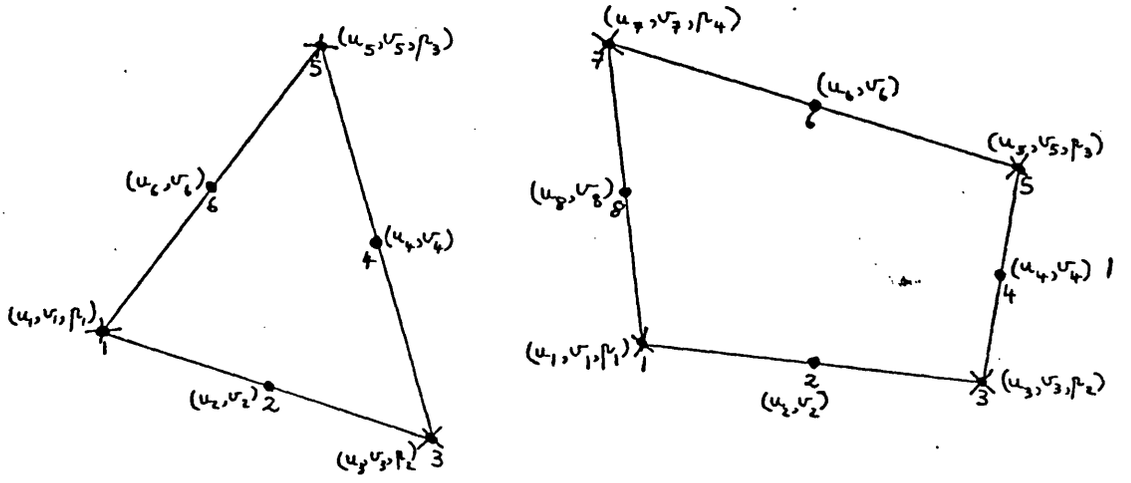


Fig. 4.3.1 Composite soil elements

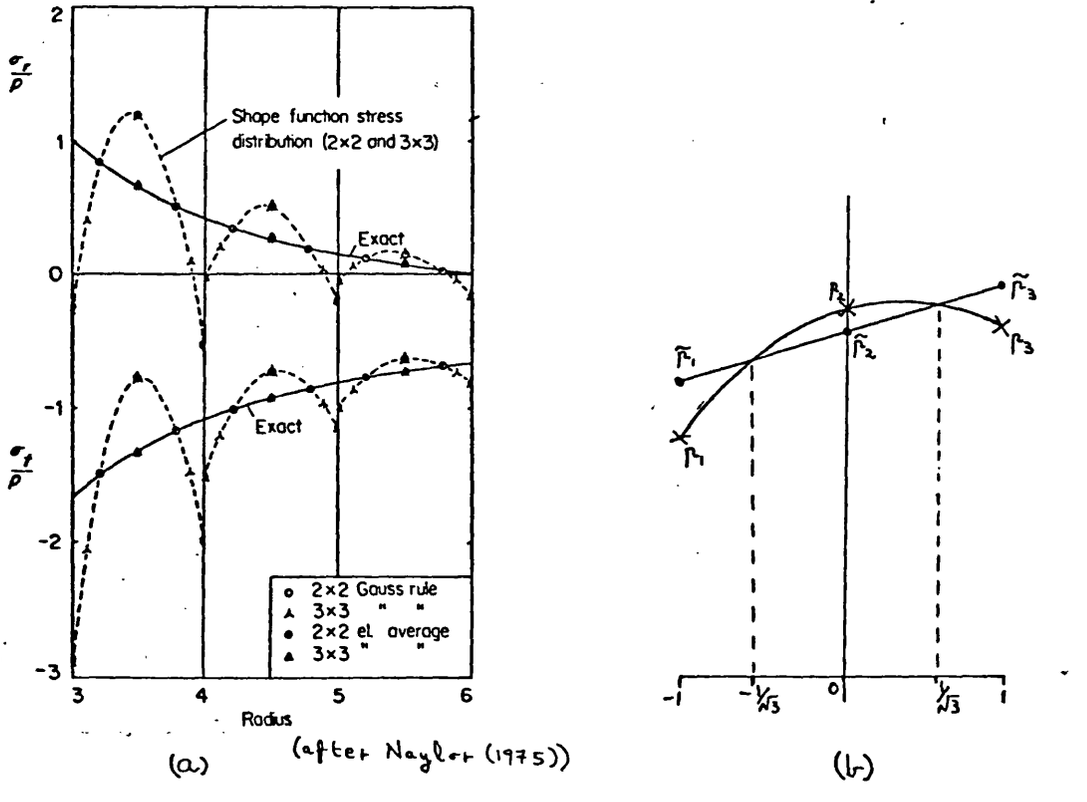


Fig. 4.3.2 Gauss-point smoothing

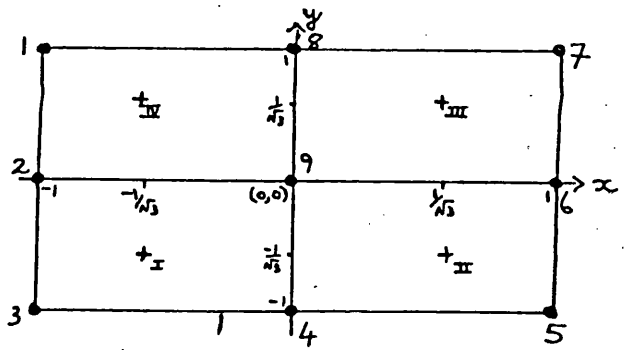


Fig. 4.3.3 Quadrilateral element with Gauss-points

$$M_1 = \begin{bmatrix} -\frac{1}{18} & \frac{1+\sqrt{3}}{9} & \frac{2+\sqrt{3}}{18} & \frac{1+\sqrt{3}}{9} & -\frac{1}{18} & \frac{1-\sqrt{3}}{9} & \frac{2-\sqrt{3}}{9} & \frac{1-\sqrt{3}}{9} & \frac{4}{9} \\ \frac{2-\sqrt{3}}{18} & \frac{1-\sqrt{3}}{9} & -\frac{1}{18} & \frac{1+\sqrt{3}}{9} & \frac{2+\sqrt{3}}{18} & \frac{1+\sqrt{3}}{9} & -\frac{1}{18} & \frac{1-\sqrt{3}}{9} & \frac{4}{9} \\ -\frac{1}{18} & \frac{1-\sqrt{3}}{9} & \frac{2-\sqrt{3}}{18} & \frac{1-\sqrt{3}}{9} & -\frac{1}{18} & \frac{1+\sqrt{3}}{9} & \frac{2+\sqrt{3}}{18} & \frac{1+\sqrt{3}}{9} & \frac{4}{9} \\ \frac{2+\sqrt{3}}{18} & \frac{1+\sqrt{3}}{9} & -\frac{1}{18} & \frac{1-\sqrt{3}}{9} & \frac{2-\sqrt{3}}{18} & \frac{1-\sqrt{3}}{9} & -\frac{1}{18} & \frac{1+\sqrt{3}}{9} & \frac{4}{9} \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{3} \\ 1 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & -\frac{1}{3} \\ 1 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{3} \\ 1 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{3} \end{bmatrix} \qquad M_3 = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M = M_3 M_2^{-1} M_1 = \begin{bmatrix} 4/9 & 4/9 & -2/9 & -2/9 & 1/9 & -2/9 & -2/9 & 4/9 & 4/9 \\ 1/9 & 4/9 & 1/9 & 1/9 & -1/18 & -2/9 & -1/18 & 1/9 & 4/9 \\ -2/9 & 4/9 & 4/9 & 4/9 & -2/9 & -2/9 & 1/9 & -2/9 & 4/9 \\ -1/18 & 1/9 & 1/9 & 4/9 & 1/9 & 1/9 & -1/18 & -2/9 & 4/9 \\ 1/9 & -2/9 & -2/9 & 4/9 & 4/9 & 4/9 & -2/9 & -2/9 & 4/9 \\ -1/18 & -2/9 & -1/18 & 1/9 & 1/9 & 4/9 & 1/9 & 1/9 & 4/9 \\ -2/9 & -2/9 & 1/9 & -2/9 & -2/9 & 4/9 & 4/9 & 4/9 & 4/9 \\ 1/9 & 1/9 & -1/18 & 1/9 & -1/18 & 1/9 & 1/9 & 4/9 & 4/9 \\ 1/36 & 1/9 & 1/36 & 1/9 & 1/36 & 1/9 & 1/36 & 1/9 & 4/9 \end{bmatrix}$$

Fig. 4.3.4 Matrices in smoothing process

4.4 Modelling the Structure

A major advantage of the finite element method applied to soil structure interaction problems, is the versatility of the method in representing structures within the global finite element assembly of soil and structural elements. We shall here detail the representation of a simple single-storey two-bay frame on footings; extension to more elaborate forms of structure and superstructure is straightforward, and can be found in standard texts such as Zienkiewicz (1977).

For a beam pin-jointed to three isolated footings, a simple finite element representation is shown in Fig.4.4.1. The joints between the bar and the triangles will act as pin-joints; should a rigid joint be required, the triangles may be replaced by rectangles having a finite area of contact with the bar. All but one of the triangle pairs might be replaced by two-noded bar elements given an arbitarily large Young's modulus to ensure their rigidity.

In a typical plane strain experiment, while the footings will extend in the out-of-plane direction to the same extent as the soil mass, the frame sturcture itself will have a much smaller out-of-plane thickness, and even have different thicknesses in different parts. This may be allowed for by adjusting the Young's modulus of each structural element so that the stiffness of the element is the same. For instance, suppose the beam in Fig.4.4.1 had an out-of-plane thickness b in the actual experiment we are modelling, a Young's modulus E and a depth h in the z-direction, while the soil and footings had an out-of-plane thickness B, as in Fig.4.4.2. Then the stiffness of the beam is given by:

$$K = \frac{EI}{l} \qquad (4.4.1)$$

where:

$$I = \frac{bh^3}{12}$$

For a beam of the same stiffness K and depth h, but with out-of-plane thickness B equal to that of the rest of the experiment, we must take an adjusted Young's modulus E*, where:

$$E^* = \left(\frac{b}{B}\right) E \qquad (4.4.2)$$

It is this adjusted Young's modulus E* which we use in the finite element program.

In program FINETIM, an inconsistency arises for nodes on the boundary between footing and soil elements; these may have pore-pressure degrees of freedom which should not appear in the element stiffness matrix formulation for the footing elements. The matter is resolved by keeping track of the type of each element: soil or structural. In dealing with structural elements, any third degree of freedom at a node is ignored, and the bookkeeping adjusted accordingly, in subroutines STIFF and FRONT.

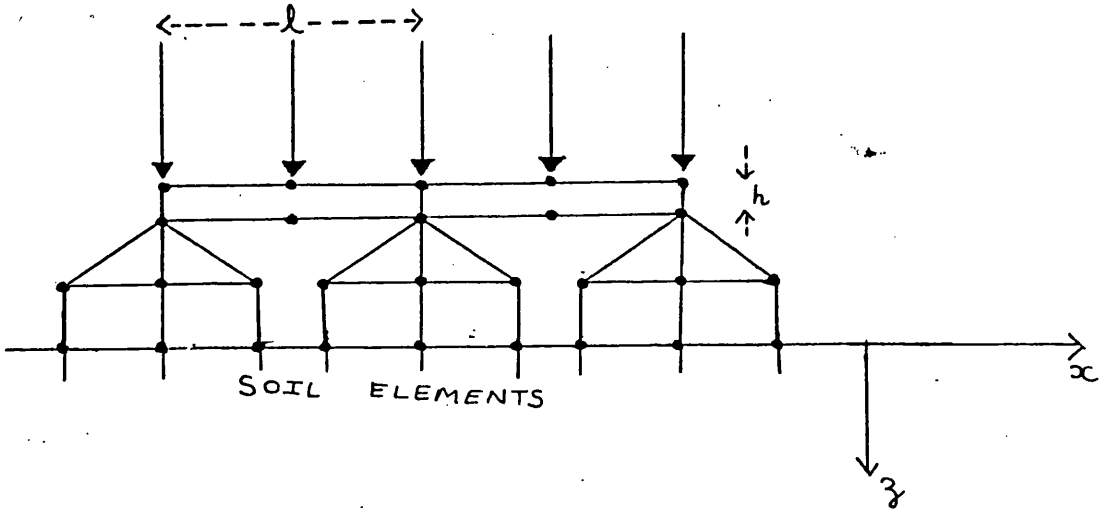


Fig. 4.4.1 Simple f.e. model of frame structure

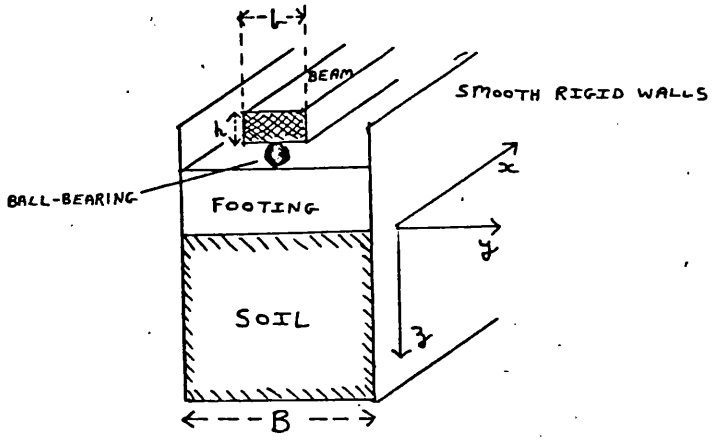


Fig. 4.4.2 Out-of-plane view of experiment

4.5 Conclusions

Although all the theory described in this Chapter has been for the case of two-dimensional plane strain problems, the extension to the other types of problem is straightforward. The main change is in the formation of the D matrix (equation 4.2.5). For three-dimensional problems, the stress vector $\underset{\sim}{\sigma}$ has six elements,

$$\underset{\sim}{\sigma} = (\sigma_x \ \sigma_y \ \sigma_z \ \tau_{xy} \ \tau_{xz} \ \tau_{yz})^T$$

and the vector of pore-pressures in equation 4.3.1 must be extended correspondingly. The program FINEPAK had been written to deal with plane strain, plane stress, axisymmetric and three-dimensional problems, and in the implementation of it and the adaption into the time-dependent form FINETIM, care has been taken to keep this generality. The only exception is the pore-pressure smoothing subroutine SMOOTH, which has not been written in a three-dimensional form.

In Chapter 6, the performance of the program FINETIM is judged against exact solutions and against that of the finite difference program FDTIM for some test problems. First, however, we will show in Chapter 5 how the two programs may be generalised to model soils which do not obey a linear elastic stress-strain law.

CHAPTER 5 : Non-linear elastic soil behaviour

5.1 Introduction

We have so far assumed that the stresses and strains in a given soil sample are related through the elasticity equations 2.3.1-3 by two simple constants: the Young's modulus E and the Poisson's ratio ν of the sample. It has long been recognized that actual soil behaviour only approaches this linear elastic ideal at low stress levels and for small strains. In practice, stresses in a soil mass are frequently increased to the point where its structure is altered and large irrecoverable strains occur.

The form that a simple stress-strain relationship actually takes, is highly dependent on the stress history of the soil. This will typically include unloading as well as loading, which produces hysteresis loops in a load/compression graph (fig 5.1.1). The types of stress-strain curve produced by such a stress history are sketched in fig 5.1.2.

The primary characteristic of stress history to affect foundation problems is the preconsolidation pressure. This is the maximum past stress the soil has experienced; should the soil be loaded past this point, a comparatively abrupt change in Young's modulus will occur, and any settlement predictions using a single value of E will be in error.

It is beyond the scope of this thesis to cover these practical problems in detail, but some aspects of the mathematical modelling involved will be considered. In §5.2 we mention graphical methods of estimating the preconsolidation pressure of a soil sample from an experimental set of pressure versus void ratio data, and propose a numerical method of performing this using smoothing splines. Models of non-linear stress-strain behaviour and their implementation in the programs FDTIM and FINTIM are discussed in §5.3, and §5.4 briefly looks at plasticity theories, which model soil behaviour in the region of the yield-point. For an excellent clear introduction to the various constitutive laws which have been developed, the reader is referred to chapter 2 of Desai and Christian (1977).

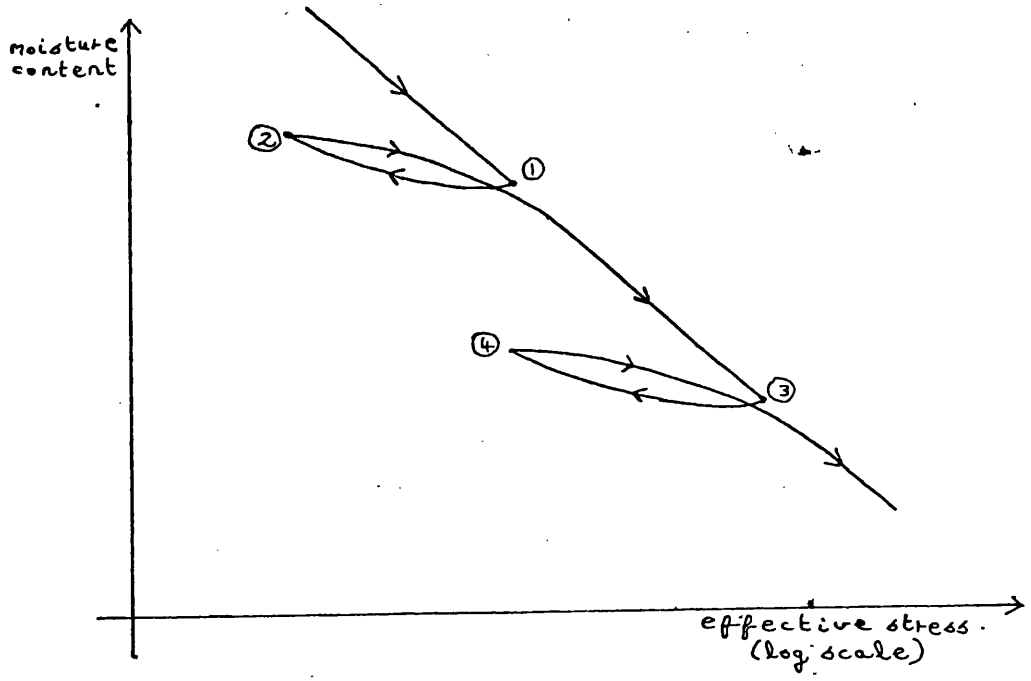


Fig. 5.1.1 Load - unload stress path

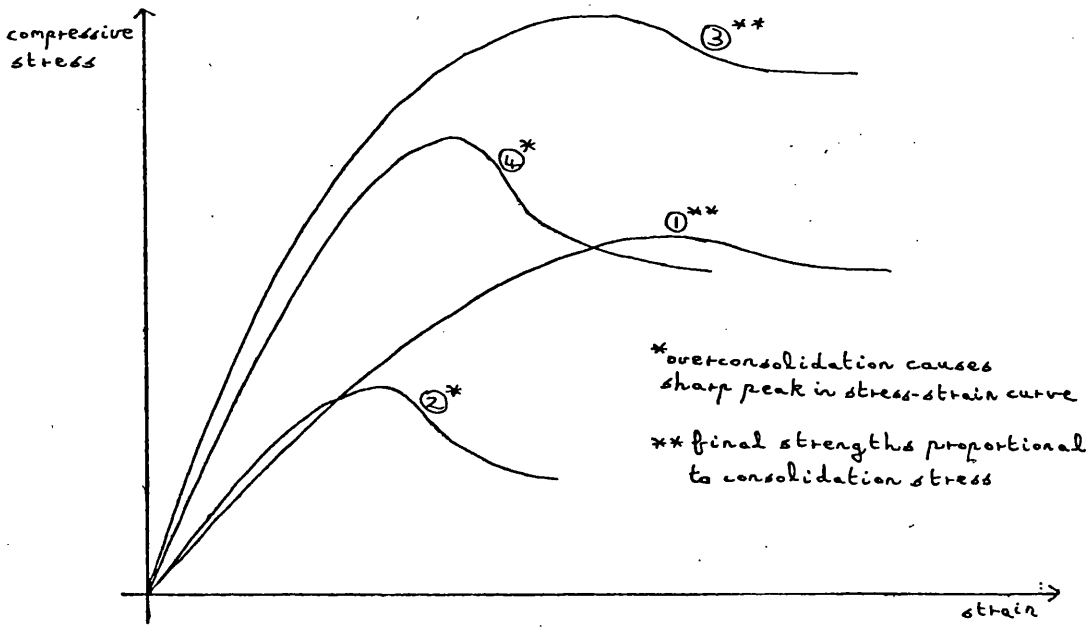


Fig. 5.1.2 Corresponding types of stress - strain curve

5.2 Estimation of preconsolidation pressure

From a one-dimensional compression test on a soil sample in an oedometer, we derive a set of data-points $\{(e_1, p'_1), \ldots, (e_k, p'_k)\}$ of void ratio e and applied effective stress p'. The geotechnical background and experimental procedure are given in soil mechanics textbooks, eg Lambe and Whitman (1969) p296. When the data are plotted on a semilogarithmic graph, an approximately straight-line relationship occurs, with an abrupt change of slope at the preconsolidation pressure p'_c of the sample. Fig 5.2.1 shows the reconstructed (straight-line) consolidation lines, and the data-points and resulting experimental curve, for a typical compression test including an unload-reload cycle. The experimental curve is usually referred to as an 'e-log p curve'. The slopes of the reconstructed reload and virgin consolidation lines are called the compression indices C_{c_1} and C_{c_2} respectively. The original in situ compression state of the sample is denoted by the point (e_o, p'_o). The laboratory test curve differs from the reconstructed lines, due mainly to disturbance of the sample during the sampling and testing procedure.

The earliest and most widely-used method for estimating the value of p'_c from such a curve was proposed by Casagrande (1936), and is illustrated in fig 5.2.2. The procedure is to:
(a) estimate the point of maximum curvature T on the curve;
(b) draw from T a horizontal line h and a tangent t to the curve;
(c) construct the line c which bisects the angle between h and t;
(d) project back from the final, straight-line portion of the curve the tangent I; the intersection of this with c gives the value of p'_c.

Although less dependent upon the user's judgement than later methods by Burmister and Schmertmann, presented in Leonards (1962), Casagrande's method does require an initial, subjective estimation of the point of maximum curvature T. It must be remembered, however, that the resulting value of p'_c is only an estimation, with a fair degree of error caused by sample disturbance as mentioned above. It is therefore reasonable to

ask if the derivation of p_c' might not be performed numerically from the data-points, avoiding the need for graphing of the e-log p curve by hand. A procedure for doing this has been developed by the writer, and programmed, tested and used on a Hewlett-Packard 9825 desktop computing system. Full details and a program specification are given in Reed (1980), but the theory involved will be outlined here.

The major problem in such a procedure is the representation of the e-log p curve itself. Let

$$x_i = \log_e p_i' \qquad i = 1, \ldots, k \tag{5.2.1}$$

$$y_i = e_i$$

to convert our variables to the normal (x, y) coordinate system. If we consider only the portion of the curve defined by increasing pressure values, ie without unloading, we have a smooth monotonically-decreasing curve asymptotic to straight lines at either end of the interval $\{x_1, x_k\}$. A most suitable model for this is the natural cubic spline

$$s(x) = a_0 + a_1 x + \sum_{i=1}^{k} c_i (x - x_i)_+^3 \tag{5.2.2}$$

where the subscript + denotes the operation

$$t_+ = \begin{cases} t \text{ if } t > 0 \\ 0 \text{ if } t \leq 0 \end{cases}$$

s(x) is a piecewise cubic polynomial function, with continuity of the first and second derivatives everywhere, while the third derivative may be discontinuous at $\{x_1, x_2, \ldots, x_k\}$; these values are called knots. Clearly, if $x < x_1$ the spline is simply a linear function, while imposition of the conditions

$$\sum_{i=1}^{k} c_i = \sum_{i=1}^{k} c_i x_i = 0 \tag{5.2.3}$$

ensures that the function is again linear for $x > x_k$. By using equations 5.2.3, together with the conditions

$$s(x_i) = y_i \qquad i = 1, \ldots, k \tag{5.2.4}$$

the coefficients a_0, a_1, c_1, c_2, ... , c_k may be determined to give a unique natural cubic spline passing exactly through the points (x_1, y_1), ... , (x_k, y_k). Since our data-points are from a laboratory test, however, they are subject to errors and the spline passing exactly through these points will generally be unnecessarily 'bumpy'; we would prefer a smooth curve passing close to the points, which is what the experimenter instinctively draws when graphing the curve by hand. Fortunately, a method of obtaining smoothing splines with this property has been proposed by Schoenberg (1964); the theory is described in Greville (1969). The degree of smoothness is determined by a positive constant g specified by the user; for a given g there is a unique natural cubic spline s(x) which minimizes the quantity

$$D = \sum_{i=1}^{k} \{s(x_i) - y_i\}^2 + g \int_{x_1}^{x_k} \{s''(x)\}^2 dx \qquad (5.2.5)$$

If $g = 0$ we obtain the spline passing exactly through (x_1, y_1), ... , (x_k, y_k), as before; a large value of g emphasises the smoothness of the curve at the expense of fidelity to the data-points, and as $g \to \infty$ s(x) tends (in the case of cubic splines) to the least-squares best straight line through the points. If the spline s(x) is written in the form of equation 5.2.2, then the coefficients are determined by equations 5.2.3 together with the conditions

$$s(x_i) + 6gc_i = y_i \qquad i = 1, \ldots, k \qquad (5.2.6)$$

replacing equations 5.2.4.

The coefficients are thus found by solving the matrix equation of order $k + 2$

$$A\underset{\sim}{c} = \underset{\sim}{b} \qquad (5.2.7)$$

where $\underset{\sim}{c} = (a_0\ a_1\ c_1\ c_2\ \ldots\ c_k)^T$

$\underset{\sim}{b} = (y_1\ y_2\ \ldots\ y_k\ 0\ 0)^T$

and the matrix A is given in fig 5.2.3. Theoretically, use of equation 5.2.2 to define s(x) may lead to the matrix A being

ill-conditioned; for the case of cubic splines with not more than twelve knots, as in a compression test, this has not been found to present any problems, and a simple Gaussian elimination solution of equation 5.2.7 is sufficient. The data-points may be individually weighted by a straightforward generalization of the theory, but this has not been found necessary.

Fig 5.2.4 shows smoothing splines for a given set of data-points; in use of the program on laboratory test results a value of $g = 0.001$ has been found consistently to give satisfactory curves.

The resulting spline curves look smooth enough to the naked eye, but as the second derivative of $s(x)$ is a piecewise linear function, with sudden changes in gradient at the knots, the point of maximum curvature always occurs in practice at a data-point. The writer has corrected for this in the program by including an empirical coefficient h in the definition of curvature $t(x)$, viz:

$$t(x) = \frac{s''}{\{1 + hs'^2\}^{3/2}} \qquad (5.2.8)$$

and use of $h = 8$, and a search algorithm to maximise $t(x)$ in the interval $\{x_1, x_k\}$, produces a point of maximum curvature in general accord with visual opinion.

The Casagrande construction may now be programmed straightforwardly. Should the initial point (e_o, p'_o) and an unload point also be known, we may use Schmertmann's (1955) assertion that a tangent from the laboratory test curve intersects with the reconstructed virgin compression line at $0.42e_o$ to determine the reconstructed consolidation lines and hence estimate the compression indices C_{c_1} and C_{c_2}.

The writer has made use of a graph-plotter peripheral to the HP9825 to produce graphical output illustrating the spline function and the constructions made. Samples of this output for actual laboratory test data are shown in fig 5.2.5. The program takes not more than twenty seconds for each data-set, and has proved very useful in analysing a large batch of tests in connection with a statistical investigation of soil properties, in lightly over-consolidated Somerset Alluvium.

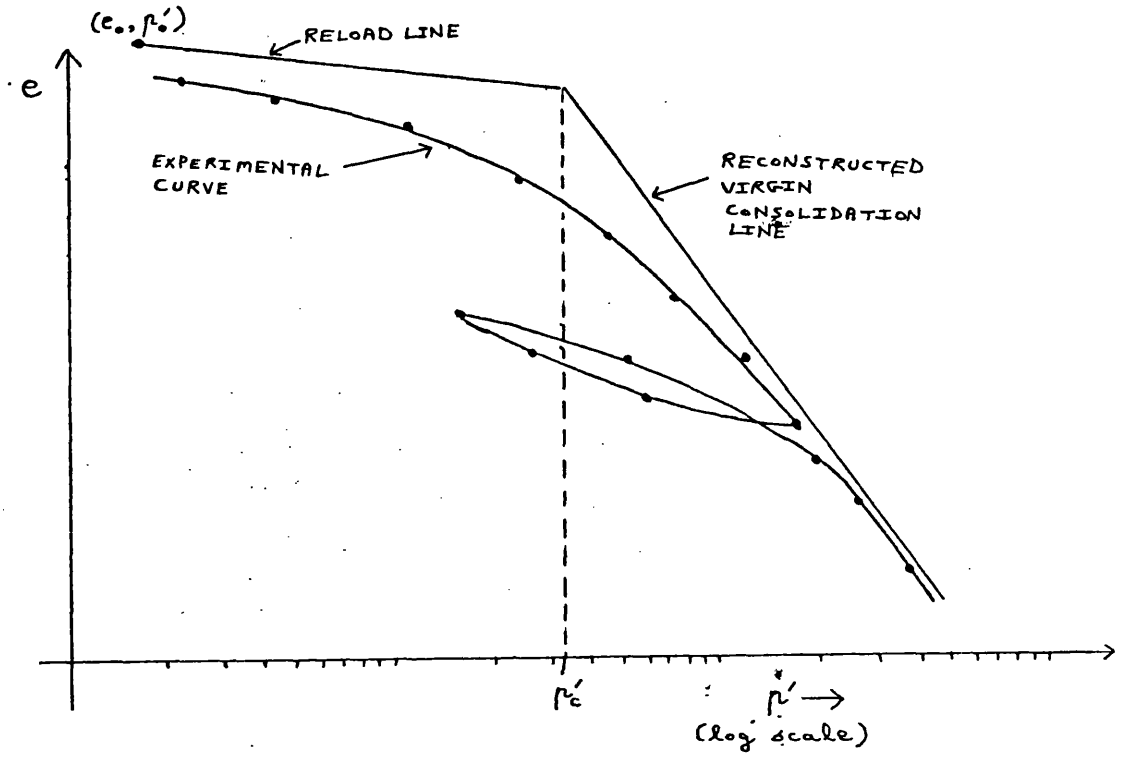


Fig. 5.2.1 Typical e - log p' curve

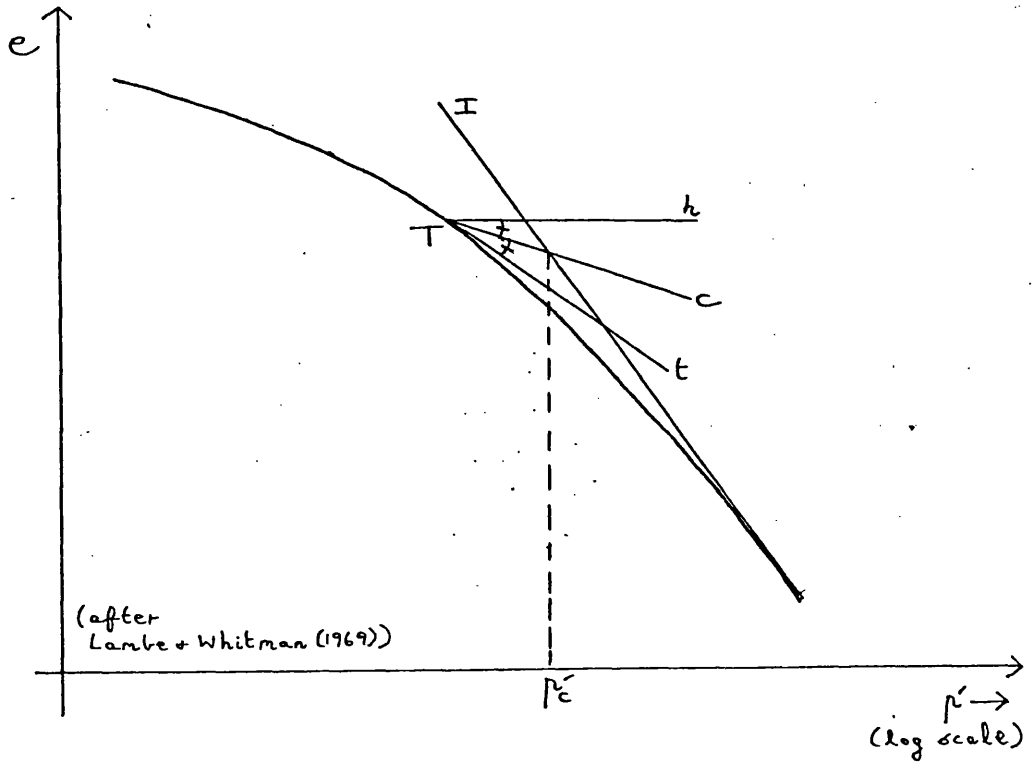


Fig. 5.2.2 Casagrande construction for p_c'

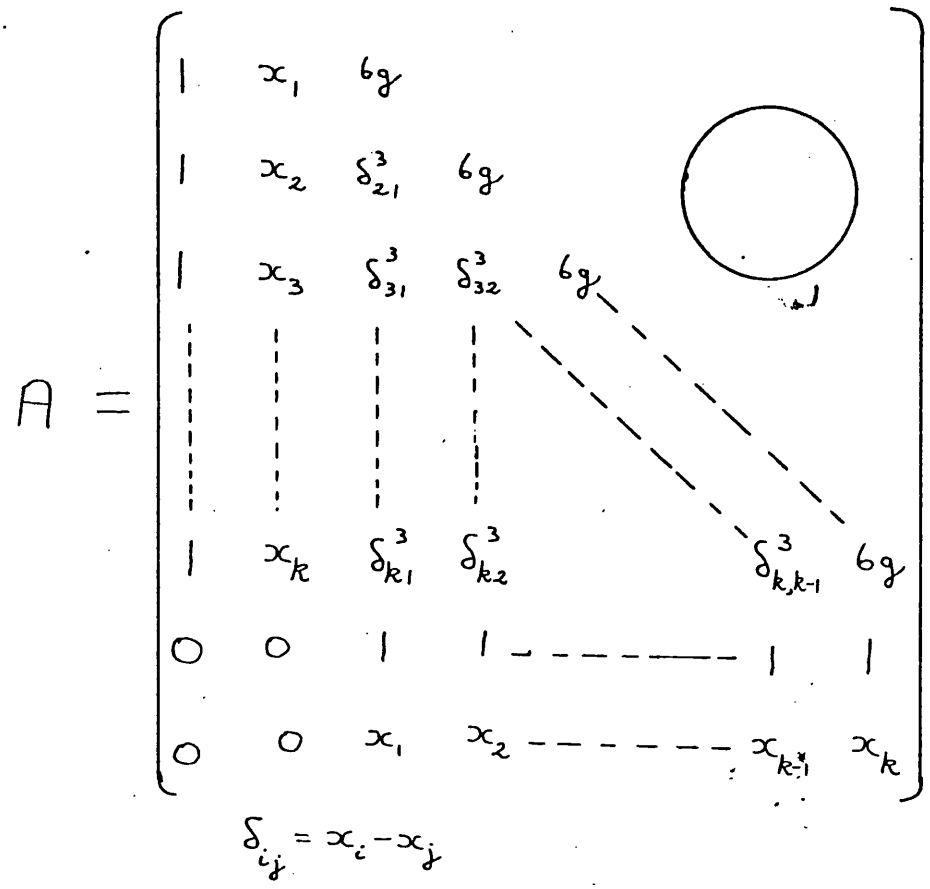


Fig. 5.2.3 Smoothing spline matrix

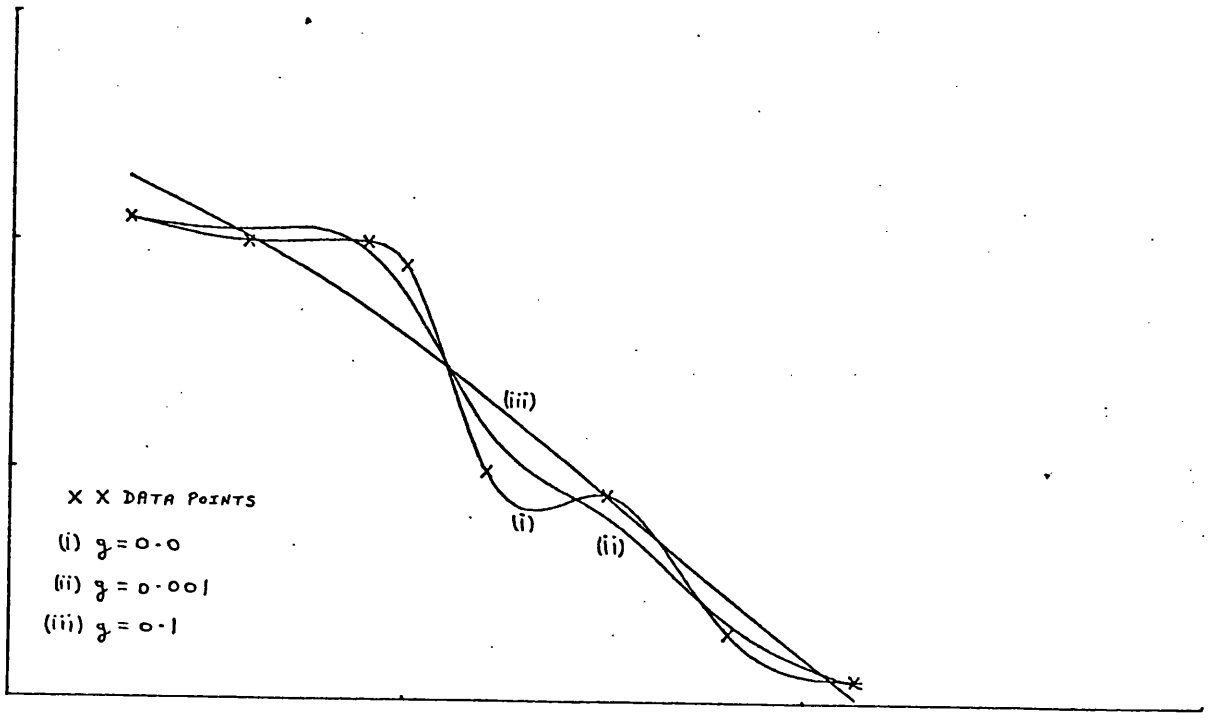


Fig. 5.2.4 Examples of smoothing splines

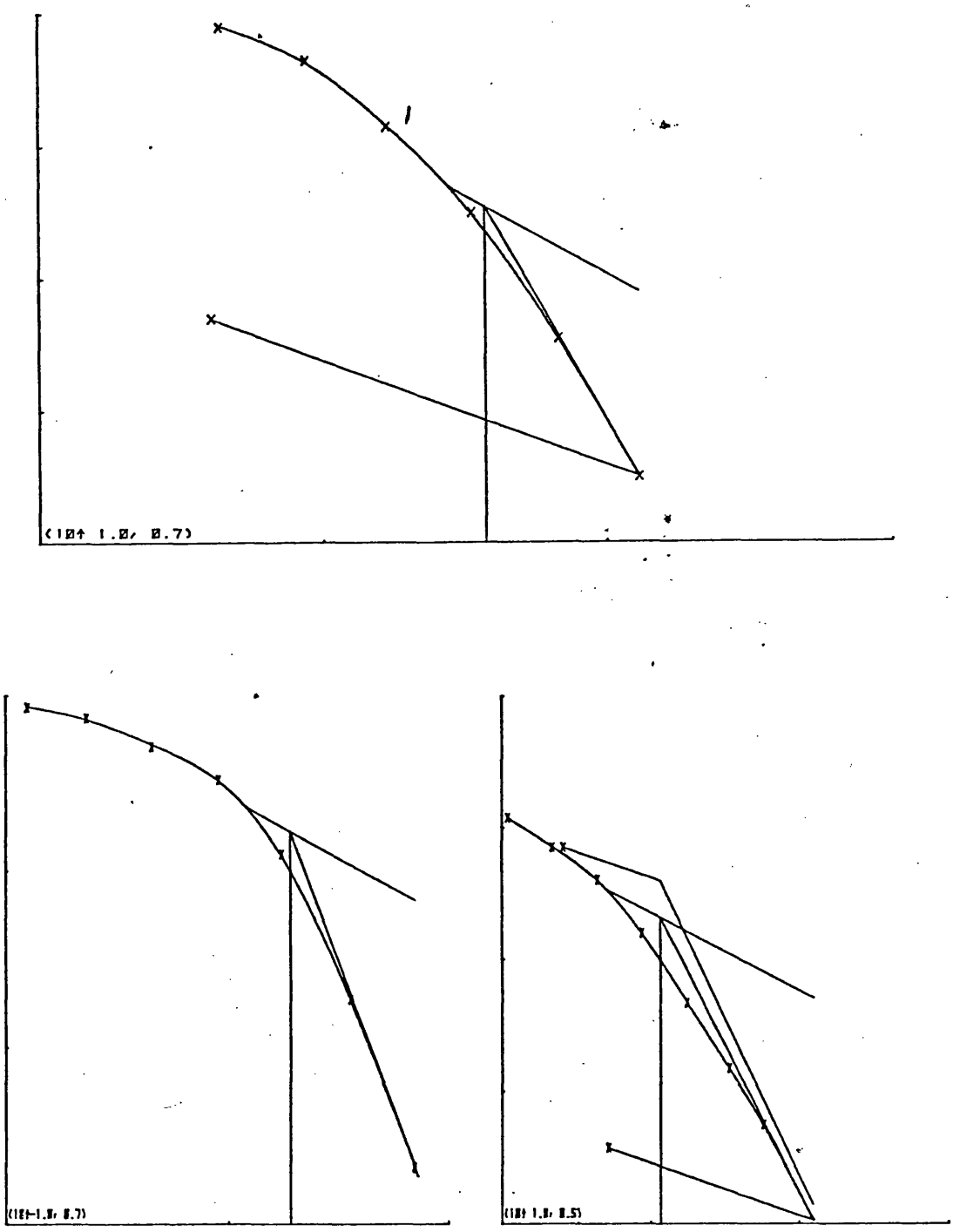


Fig. 5.2.5 Graphical output from program to find p_c'

5.3 Non-linear elastic models

The simplest alteration that can be made to the linear elastic law in order to better represent the true stress-strain behaviour of soils, is the bilinear law. This uses an initial Young's Modulus E_i until the deviator stress $\sigma_1 - \sigma_3$ exceeds a critical value, when a smaller 'yielded' modulus E_y is used. (Here and in all that follows, effective stresses and stress parameters are assumed.) Fig 5.3.1 shows that this is equivalent to modelling the true stress-strain curve by a pair of straight lines. The choice of E_i, E_y and the yield-point is fairly arbitrary, however, as fig 5.3.1 shows. By going one step further we may construct a multilinear model (fig 5.3.2), using a number of straight-line curves. This is particularly useful if we are determining the constitutive law from a set of results $(\varepsilon_1, \sigma_1), \ldots, (\varepsilon_k, \sigma_k)$ from a triaxial test on a soil sample. Also of use in this situation is the model originated by Desai (1971) using a cubic spline to construct a smooth curve through the data-points. For any given stress, we differentiate the spline function to obtain the tangent modulus E_t at that point. It is generally accepted, however, that laboratory tests to determine the modulus of a soil sample will seriously underestimate the in situ value. This again is a problem of disturbance and possibly sample size.

In the absence of a detailed stress-strain curve, the models described so far employ a number of coefficients which have no physical meaning. The user is much happier with a model which can take as input familiar, measurable physical quantities such as the cohesion c and angle of friction ϕ for the soil. Two such models will now be outlined: the hyperbolic model and the K-G model.

Hyperbolic model

The hyperbolic model was developed by Duncan and Chang (1970) from Kondner's (1963) finding that a plot of deviator stress $\sigma = \sigma_1 - \sigma_3$ against axial strain ε in a triaxial compression test is very nearly a hyperbola of the form

$$\sigma = \frac{\varepsilon}{b + a\varepsilon} \qquad (5.3.1)$$

where a and b are constants, which may be related to engineering properties as follows.

For very small ε,

$$\sigma \approx \frac{\varepsilon}{b} \qquad (5.3.2)$$

so that $1/b$ is the initial slope of the curve, ie the initial Young's modulus E_i. For large strains,

$$\sigma \approx\approx \frac{1}{a} \qquad (5.3.3)$$

so that $\sigma = 1/a$ is the asymptote to the curve. In practice the stress-strain curve will deviate from the hyperbolic model at large strains, reaching its (ultimate) compressive strength s and yielding before it has approached the asymptote; it is therefore usual to include a corrective constant R_f, where $0.7 < R_f < 0.9$, so that we write

$$s = R_f \cdot 1/a \qquad (5.3.4)$$

The meanings of $1/a$ and $1/b$ are shown graphically in fig 5.3.3. Substituting for a and b in equation 5.3.1, and differentiating to find the tangent modulus E_t as a function of σ,

$$E_t = \frac{\partial\sigma}{\partial\varepsilon} = \frac{b}{(b + a\varepsilon)^2} = \frac{1}{b}(1 - a\sigma)^2 = E_i\left(1 - \frac{R_f\sigma}{s}\right)^2 \qquad (5.3.5)$$

The compressive strength s of a particular soil depends upon the all-round stress σ_3 acting; for a soil with a Mohr-Coulomb failure criterion (see §5.4) this relationship can be expressed

$$s = \frac{2\sigma_3\sin\phi + 2c\cos\phi}{1 - \sin\phi} \qquad (5.3.6)$$

Substituting into equation 5.3.5 yields the following formula for tangent modulus $E_t = E_t(\sigma_1, \sigma_3)$:

$$E_t = E_i\left\{1 - \frac{R_f(\sigma_1 - \sigma_3)(1 - \sin\phi)}{2\sigma_3\sin\phi + 2c\cos\phi}\right\}^2 \qquad (5.3.7)$$

There is an empirical formula for E_i in terms of σ_3 (see Desai and Christian (1977), p84), but as this introduces new unknown coefficients without physical meaning, we will not include it here. Wood (1972) has extended the model to the three-dimensional case, using the von Mises failure criterion.

K-G model

The models so far described have not involved any mention of the Poisson's ratio ν of the soil; this has been assumed to remain constant throughout compression. Such an assumption is hardly warranted, and the following model, described in Naylor (1978), lecture 4 from which the following outline is derived, avoids this.

Instead of taking Young's modulus and Poisson's ratio as the primary stress-dependent soil properties, we consider the bulk modulus K and shear modulus G, related to E and ν by

$$K = \frac{E}{3(1-2\nu)}$$
$$G = \frac{E}{2(1+\nu)} \qquad (5.3.8)$$

Suppose that K and G depend linearly upon the stress state in the form

$$K = K_1 + a_k\sigma_s$$
$$G = G_1 + a_g\sigma_s + b_g\sigma_d \qquad (5.3.9)$$

where $\sigma_s = \frac{1}{2}(\sigma_1 + \sigma_3)$ and $\sigma_d = \sigma_1 - \sigma_3$, denoting the spherical and deviatoric stress invariants respectively. There are five unknown coefficients. Application of the Mohr-Coulomb failure criterion allows two of these to be eliminated, through the relations

$$b_g = -0.5\, a_g \operatorname{cosec}\phi$$
$$G_1 = a_g\, c \cot\phi \qquad (5.3.10)$$

and two may be deduced from an e-log p curve:

$$K_1 = 0$$

$$a_k = \frac{1+e}{C_c} \log_e 10 \qquad (5.3.11)$$

where C_c is the compression index appropriate to the current stress state (ie dependent upon whether virgin consolidation or reloading is taking place - see §5.2). The remaining constant a_g is found from a measurement of undrained shear strength C_u and initial shear modulus G_i. Although a better model of the physical changes occurring in the soil during compression, the K-G model requires a greater knowledge of soil behaviour in laboratory tests than does the hyperbolic model; it has a further disadvantage in terms of implementation in a finite element model of consolidation, as will be explained in the following section.

Implementation

Inclusion of a non-linear elastic constitutive law in the finite difference program FDTIM is very simple; at each stage when vertical strains ε_z are being evaluated at the mesh-points, a subroutine is called for each mesh-point to determine the current tangent modulus E_t, a function of the principal stresses σ_1 and σ_3 (found from σ_x and σ_z by equation 2.4.12) then acting. In the case of the K-G model, both E_t and ν must be determined; these are found from K and G by transforming equations 5.3.8 into

$$E = \frac{9KG}{3K+G}$$

$$\nu = \frac{3K-2G}{2(3K+G)} \qquad (5.3.12)$$

In a finite element analysis, the nodal stresses must be determined after the stiffness matrix equation has been solved for the displacements, by

$$\underset{\sim}{\sigma} = DB\underset{\sim}{\delta} \qquad (5.3.13)$$

the D and B matrices being as defined in §4.2. In FINEPAK this is done in subroutine OUTPUT. Because of the large stress

change, one must either apply the load in a number of increments, changing the modulus after each increment (the incremental method), or iterate using in each resolution the value of E determined from the previous one until compatible displacements and E-values are found (the iterative method). Ideally, both methods should be combined, iterating to convergence for each increment. Such procedures take time and money, of course; Hagmann (1971) has investigated the relative efficiency of the two approaches, and concluded that most benefit was obtained from increasing the number of load increments. In the case of a time-dependent analysis, this application of the load in increments occurs automatically with the timestepping, through the mechanism of the dissipation of excess pore-water pressure.

Since each element D matrix is a function of Young's modulus, as defined in equation 4.2.5, it would appear to be necessary to recalculate the element stiffness matrices by calling subroutine STIFF for each resolution, in the time-dependent case at each timestep. However, since E only appears as a factor multiplying each element of the matrix, it is possible to avoid this by calling STIFF initially using a value of $E = 1$, storing the element stiffness matrices on temporary file, and then at each resolution multiplying the element stiffness matrices by the current E value during assembly in FRONT. This strategy has been employed in FINTIM, and extended also to the factor of $(1 - \theta)\Delta t$ multiplying submatrix H - see §4.3. This time-saving scheme cannot be employed with the K-G model, since ν as well as E is altered at each resolution.

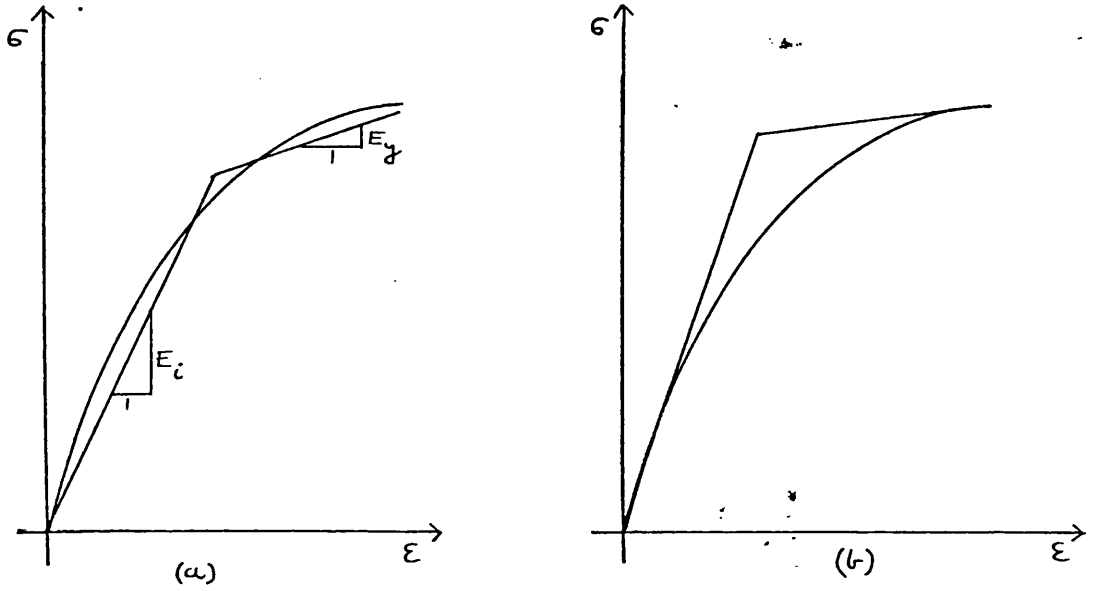


Fig. 5.3.1 Examples of bilinear models

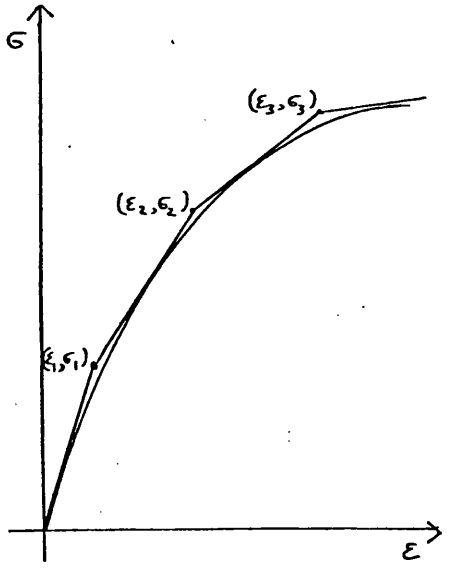


Fig. 5.3.2 Multilinear model

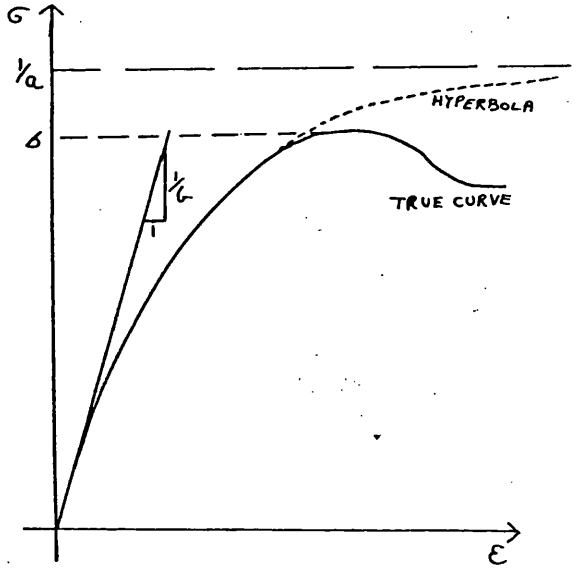


Fig. 5.3.3 Hyperbolic model

5.4 Failure Criteria

The Mohr-Coulomb failure criterion for a cohesive soil is illustrated in fig 5.4.1. If the minor effective principal stress σ_3 is held constant, and σ_1 increased, failure occurs when the Mohr's circle of stress reaches the failure line, as shown, ie when

$$\sin\phi = \left(\frac{\sigma_1 - \sigma_3}{2}\right) / \left(c\cot\phi + \frac{\sigma_1 + \sigma_3}{2}\right) \qquad (5.4.1)$$

which may be rearranged to the form given in equation 5.3.6, with $\sigma_1 - \sigma_3$ at failure denoted by the compressive strength s. The Mohr-Coulomb criterion is independent of the intermediate effective principal stress σ_2, and this creates sharp 'corners' when the criterion is sketched in three-dimensional stress space. This drawback does not occur with the extended von Mises criterion, which for a cohesionless soil is

$$(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 = 2\alpha^2\left\{\frac{\sigma_1 + \sigma_2 + \sigma_3}{3}\right\}^2 \qquad (5.4.2)$$

Work by a number of researchers over the past twenty years, summarized in Wood (1972), p15, indicates that while the Mohr-Coulomb criterion is the most suitable for sands, the extended von Mises criterion is more appropriate as a model of the behaviour of clays. In general, a normally consolidated clay has $c = 0$, and the value of the cohesion increases with increasing over-consolidation ratio.

All the models so far described are essentially valid only when the soil is not close to failure, and they all assume a direct relationship between stress and strain. There is no consideration of the stress path followed by the soil in reaching its current stress state. These defects (for stress history may in practice have a considerable influence on soil properties) are avoided in models incorporating plasticity and viscosity. Consideration of these models lies outside the scope of this thesis, but some basic definitions will be made here for completeness since this is a field of considerable current interest. For further details the reader is referred to Desai and Christian (1977), chapters 2 and 3, and Naylor (1978), lecture 3.

In an elasto-plastic model a failure criterion such as the ones described above is incorporated in a yield function $F(\underset{\sim}{\sigma}, \underset{\sim}{\varepsilon})$. The yield surface is defined by

$$F(\underset{\sim}{\sigma}, \underset{\sim}{\varepsilon}) = 0 \qquad (5.4.3)$$

and when the soil reaches the yield surface plastic (ie irreversible) strains occur. The differential increments of strain in the coordinate directions are usually taken to be proportional to the outward normals to the yield surface at the point of contact; this condition defines the associated flow rule. Inclusion of strains in the definition of F enables strain-hardening behaviour to be modelled.

In the elasto-plastic model the yield surface cannot be exceeded; in visco-plasticity this may happen, and in this situation creep strains develop. Creep strains are frequently observed in practice, and so visco-plasticity has become very popular as a realistic soil model. The time-dependent nature of creep strain means that a visco-plastic or visco-elastic model is especially suited to a consolidation analysis, and Booker and Small (1977) have developed the finite element technique for this.

We have assumed that when the soil is inside the yield surface it behaves purely elastically. It is usually the case, however, that if any load is applied to a soil and later removed the soil does not completely return to its previous volume, ie plastic strains will develop even well inside the failure envelope. This has led to the development of so-called 'capped' yield models, where the failure envelope is closed with a cap which expands outward with the stress path to model strain hardening. Such a model may be designed to incorporate the critical state concepts developed by Roscoe and his colleagues and explained in Schofield and Wroth (1968); this is detailed in Zienkiewicz and Naylor (1972) and in lecture 5 of Naylor (1978).

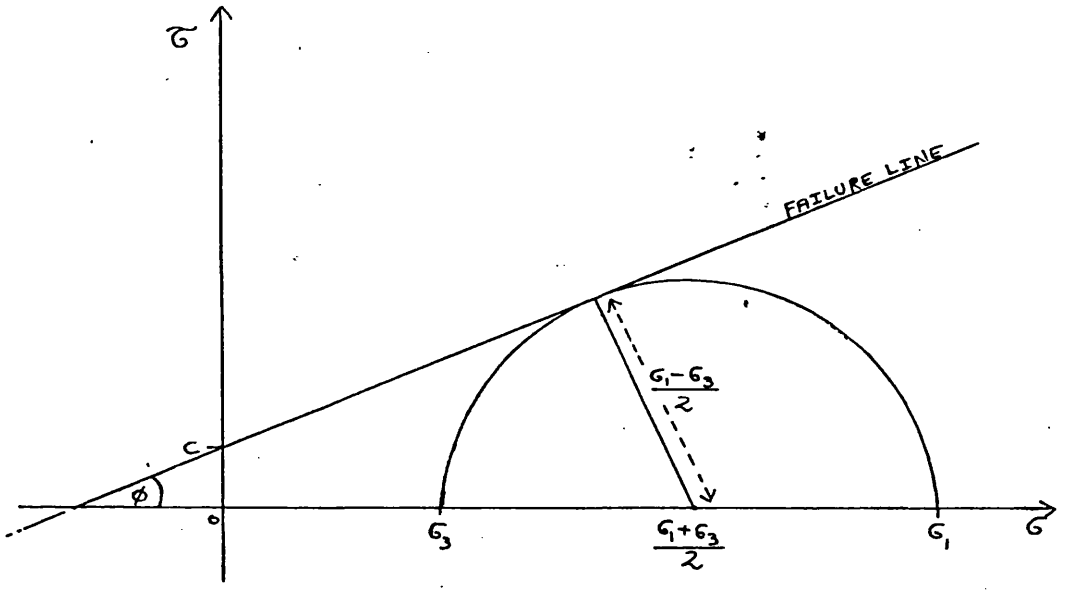


Fig. 5.4.1 Mohr - Coulomb failure criterion

5.5 Conclusion : choice of model

Various non-linear soil models have been discussed. For the problem in hand, where we are primarily interested in modelling the soil-structure interaction effects, use of an elasto-plastic or visco-plastic constitutive law is not warranted at this stage of research. The most attractive direct stress-strain laws are the hyperbolic and the K-G models, and one must weigh the computational advantage of the hyperbolic law against the physically more realistic K-G model. As the K-G model requires examination of a triaxial compression test curve for its parameters, while the hyperbolic model requires only the values of c and ϕ, it is considered that on balance the hyperbolic law is more suited to the present, mainly theoretical investigation.

A problem not so far mentioned directly, but which is more important in the present situation than in ordinary consolidation, is that of load reversal. During consolidation the structural load is continuously redistributed across the footings, so that the soil under a particular footing may experience a succession of partial unloads and reloads during the consolidation process.

A very simple and neat way of adapting the hyperbolic law to allow plastic strains to develop during this process, is to stipulate that when an unload occurs (ie when the deviator stress $\sigma_1 - \sigma_3$ decreases) the Young's modulus used will be not the tangent modulus E_t but the initial modulus E_i. The effect of this on the stress-strain curve is shown in fig 5.5.1.

This adapted hyperbolic law has been included in the programs FDTIM and FINETIM.

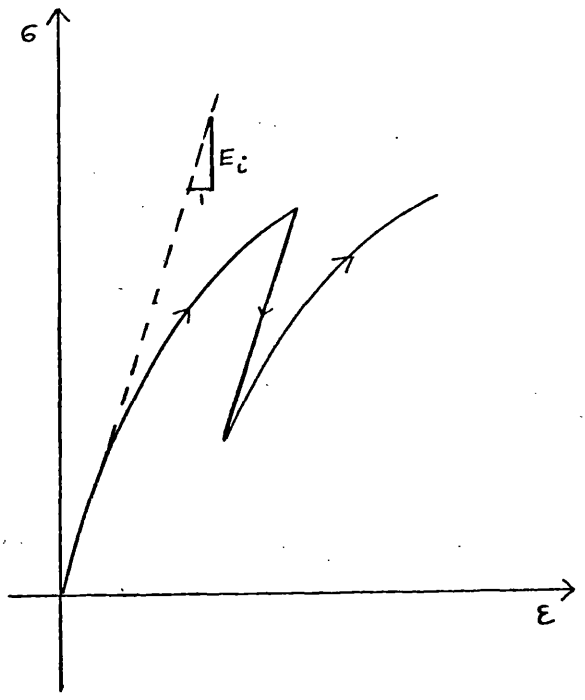


Fig. 5.5.1 Simplest description of unload modulus

CHAPTER 6 Results (i): flexible and rigid footings

6.1 Introduction

We here illustrate the methods which have been developed in the preceding chapters by using the programs FDTIM and FINETIM to model single-footing plane strain problems. The consolidation settlement of a flexible footing on a homogeneous, isotropic linear elastic soil in plane strain may be solved analytically. Results for a soil of infinite depth are given in Schiffman, Chen and Jórdan (1969), and for a finite soil layer on a smooth impervious base by Gibson, Schiffman and Pu (1970). Booker (1973) solved the problem analytically for a rough impervious base, and obtained good results from a finite element analysis, rather poorer results from a finite difference program using Burmister's (1956) stress distribution formulae. In §6.2 we compare the predictions of both FINETIM and FDTIM to analytic solutions for both infinite-depth and smooth-base problems.

For the case of a rigid footing, no analytic solutions exist and an attempt was made to obtain time-settlement curves from a model footing test, to compare with numerical solutions. The results are presented in §6.3.

Not shown here are checks which were made on the time-dependent finite element program FINETIM, to ensure that for large times the predicted soil deformations approached those of the drained analysis of Naylor's program FINEPAK.

6.2 Flexible footings

Our programs FINETIM and FDTIM are designed to model plane strain consolidation of a soil layer. An analytic solution for the surface settlement with time for this problem, with a uniform strip-load and a smooth impervious base, has been published by Gibson, Schiffman and Pu (1970). In their paper they published graphs prepared from this solution of centreline settlement in the case of a soil layer of depth equal to the strip-load half-width. These were used by Hwang, Morgenstern and Murray (1971) as a check on the accuracy of their finite element program (which employed a logarithmic timestepping scheme), and the writer has followed this example. Hwang et al obtained exceptionally good agreement with Gibson's curves, but they did not give any details of the finite element mesh used, etc, in their paper.

Finite Element Program FINETIM

The writer's finite element mesh is shown in fig 6.2.1. It comprises forty eight-noded composite quadrilateral elements (see fig 4.3.1(b)), with a total of 149 nodes. As the analytic solution is for a soil layer extending infinitely in the x-dimension, the columns of elements at either end of the mesh were lengthened until the end-conditions no longer affected the centreline displacements; this was found to occur when the ends were ten times the strip-load half-width away from the centreline.

Timestepping was arranged with an initial timestep of 2, and was increased by a factor of 10 after every five steps; this achieved roughly evenly-spaced points on a logarithmic plot of the time factor. Results were obtained and compared with Gibson's curves for the cases $\nu = 0.0$ and $\nu = 0.2$; these are presented in fig 6.2.2. The vertical axis is of the dimensionless settlement factor

$$\delta = \frac{Gw}{ap} \qquad (6.2.1)$$

where w = centreline displacement

a = strip-load half-width

p = strip-load intensity

and $G = \frac{E}{2(1+\nu)}$ = shear modulus of the soil.

The horizontal axis is a logarithmic plot of the adjusted time factor

$$\tau = 2G\frac{k}{\gamma_w}\frac{1}{D^2}t \qquad (6.2.2)$$

where k = soil permeability

γ_w = water density

D = layer depth

and t = elapsed time.

There is reasonable agreement between the analytic and finite element solutions. The largest errors occur in the initial settlements, as was reported by Sandhu et al (1977) - see §4.3. A small oscillation in the results is noticeable at the largest timestep. The early stages of consolidation were also modelled for the case $\nu = 0.0$ using the same mesh, but normal eight-noded quadrilateral elements and employing the Gauss-point smoothing subroutine SMOOTH to remove the initial oscillations of pore-pressure. The theory of this process is described in §4.3.

Figure 6.2.3 shows the initial nodal pore-pressures down the centreline before and after smoothing, as well as the initial pore-pressures given by the composite elements. While the initial unsmoothed pore-pressures are widely scattered, the smoothing process produces a very reasonable pore-pressure curve. The composite element pore-pressures are greater than the smoothed ones (by as much as 80%), and show some sign of an oscillation themselves. The initial centreline displacement factor δ is 0.259 for the quadratic-elements-with-smoothing model, as against 0.249 for the composite-element model and about 0.257 for the exact solution.

As far as the writer is aware, there are as yet no detailed solutions published for the excess pore-pressure distribution

in a finite soil layer during plane strain consolidation. The case of an infinite half-space of soil under a unit strip-load of half-width a, however, has been systematically analysed by Schiffman, Chen and Jordan (1969). They were particularly interested in the different predictions of the 'true' Biot consolidation theory and the 'pseudo' Terzaghi-Rendulic theory (see §3.2). Hwang et al used the Biot theory pore-pressure curves from this paper as a check on their finite element program, and reported achieving good agreement by representing the half-space by a soil body of width 12a and depth 9a. The writer has followed this example, and constructed a model with the mesh shown in fig 6.2.4, having 56 elements and 199 nodes.

Schiffman et al published graphs of the distribution of pore-pressure at a time after loading corresponding to an adjusted time factor τ of 0.1. The distribution of pore-pressure with depth down the centreline is shown in fig 6.2.5, and along the horizontal plane $x = 0.5a$ in fig 6.2.6. Curves from both the Biot and the Terzaghi-Rendulic theories were given by Schiffman et al; the latter results were subsequently criticized by Viggiani, Davis and Poulos (1970) as being based upon an incorrect definition of the coefficient of consolidation in plane strain, and so only the Biot theory curves have been reproduced here. We also show the nodal pore-pressures resulting from using a) composite elements, and b) quadratic elements with smoothing.

The smoothed pore-pressures appear to give a more accurate approximation to the theoretical curves generally, except very close to the centre of the loaded area, where the pore-pressure is underestimated. It is to be expected that the smoothing process is least effective in this region where the pore-pressure distribution varies most rapidly and there are prescribed values of pore-pressure on the permeable boundary. A finer mesh would improve the accuracy of the results.

The approximate times taken in running the program for the two models on an ICL 2980 virtual-store computer are as follows. Using composite elements in the mesh (ie a total of 478 nodal

variables), the subroutines INPUT, STIFF, LOAD, FRONT and OUTPUT for the initial settlements took 35 seconds, and each subsequent time increment took 10 secs.

Using normal quadratic elements (ie with 597 nodal variables) solution for the initial unsmoothed settlements and pore-pressures took 41 secs, the smoothing process and resolution took 15 secs, and each subsequent timestep took 13 secs. Compared to using composite elements, the use of quadratic-elements-with-smoothing on the same mesh therefore needs about 20% more computer time. This is caused by the extra degree of freedom at mid-side nodes, plus a time equivalent to slightly more than that for one timestep for the smoothing procedure itself. Against this extra cost, it should be remembered that to obtain the same amount of nodal pore-pressure data with composite elements would require a finer mesh having four times as many elements.

The efficiency of the program compares well with Hwang et al's reported run-times for their finite element program on an IBM 360-67 computer, of approximately 120 secs for the initial solution and 20 secs per subsequent timestep on a problem with 420 nodal variables.

Finite Difference program FDTIM

The finite layer consolidation problem solved by Gibson, Schiffman and Pu (1970) and described above, was also used by the writer to test the accuracy of the finite difference program FDTIM. The dimensions of the model are the same as in fig 6.2.1, and a mesh of 25 columns and 11 rows was used. The program was run using both the infinite-depth stress evaluation subroutine BOUSS, and the finite-layer stress subroutine FILON (see §2.2), and the resulting graphs of centreline settlement with time are compared with Gibson et al's theoretical curves in fig 6.2.7. The Boussinesq stress formulae prove to be invalid in this problem, while use of the finite-layer stresses gives accurate predictions of settlement for most of the consolidation process, although final drained

settlements are underestimated by slightly less than 4%.

With a total of 275 meshpoints, program FDTIM took 44 seconds to run on the ICL 2980, about a quarter of the time needed (192 seconds) for solution of the same problem with the finite element program FINETIM using the mesh in fig 6.2.1 (149 nodes). In each case 25 timesteps were taken.

The writer has also analysed using FDTIM, the problem of Schiffman et al, of the excess pore-pressure distribution in a half-space during plane-strain consolidation. Results from the finite element program have already been compared with the Biot solution in figs 6.2.5-6. A mesh of 25 columns and 17 rows was used to cover the same region as in the finite element analysis (see fig 6.2.4), and the same timestepping scheme was employed. The stress distribution was found using the Boussinesq infinite-depth formulae as contained in subroutine BOUSS. Figs 6.2.8-9 show graphs of the excess pore-pressure distribution down the centreline and at a depth of 0.5a, at a time equal to a time factor τ of 0.1, and in both cases there is an equally good correspondence with the theoretical solution as was obtained by the finite element analysis. The pore-pressures down the centreline are slightly less than those predicted by Biot theory, causing slightly greater centreline immediate settlement (0.83 from FDTIM, as against 0.79 and 0.81 from FINETIM using composite and quadratic elements respectively). The program took 15 secs to perform the initial undrained analysis, and then 2¼ secs for each timestep. Compared to the finite element model with composite elements, this is half the time needed for the initial solution, and a quarter of the time needed for each timestep.

The times taken by the program FDTIM for this problem could be further halved by making use of the symmetry of the problem to model only one half of it, placing an impermeable boundary along the centreline. This is done in the following section.

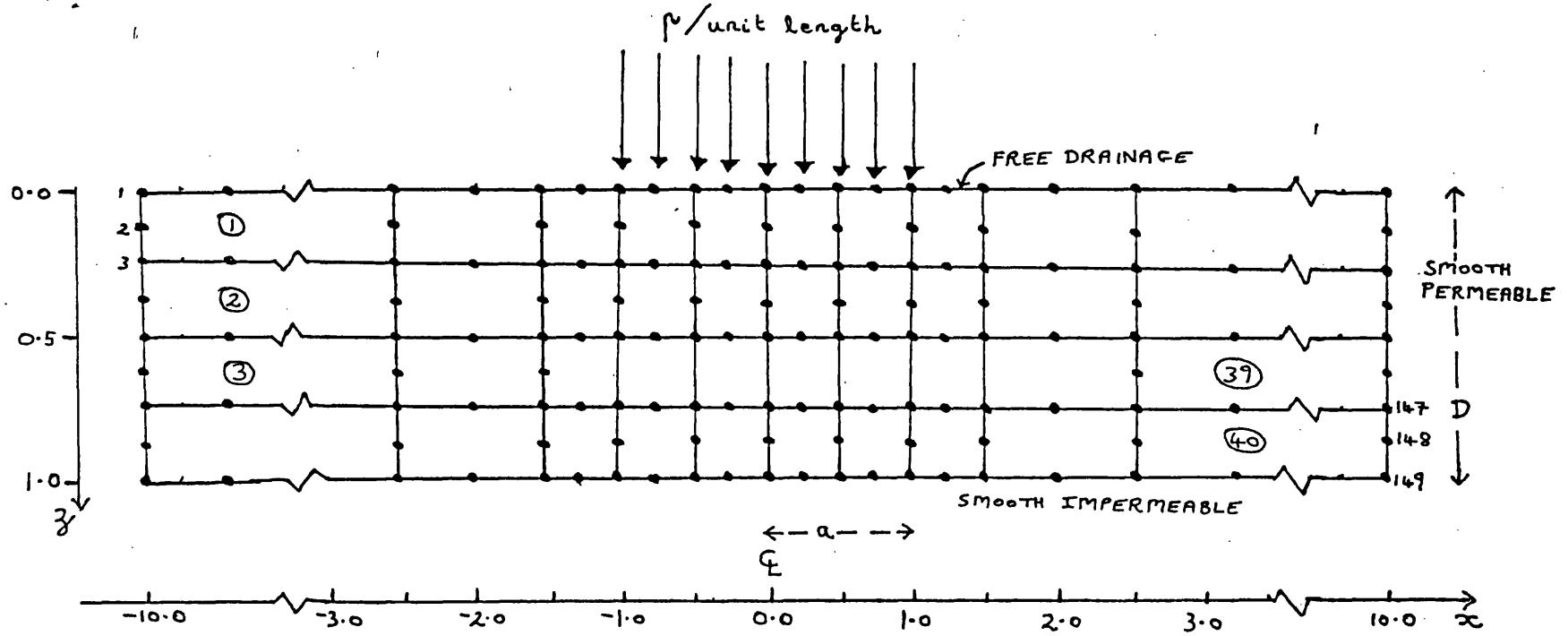


Fig. 6.2.1 Finite element mesh for finite-depth problem

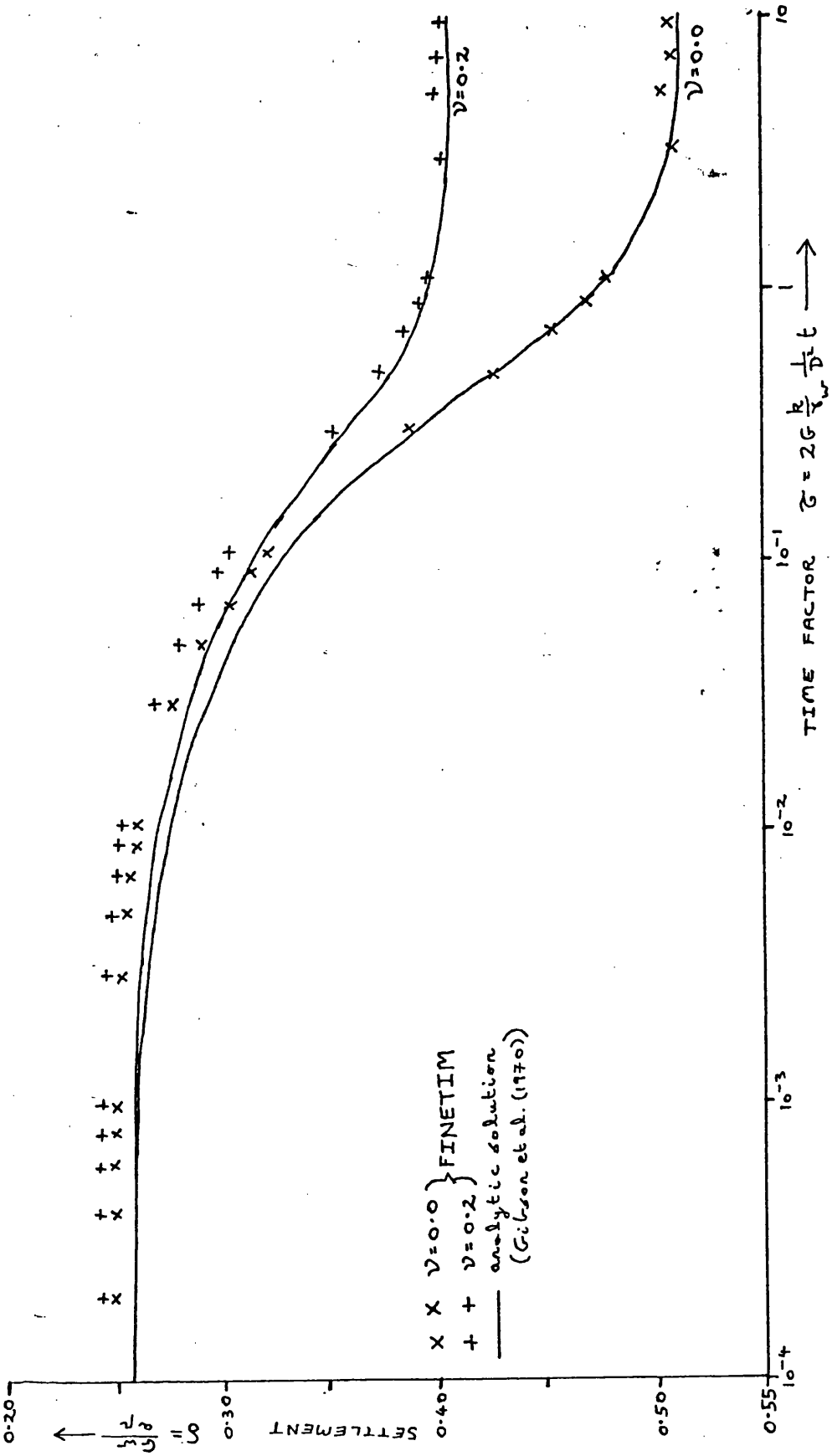


Fig. 6.2.2 FINETIM time - settlement curves

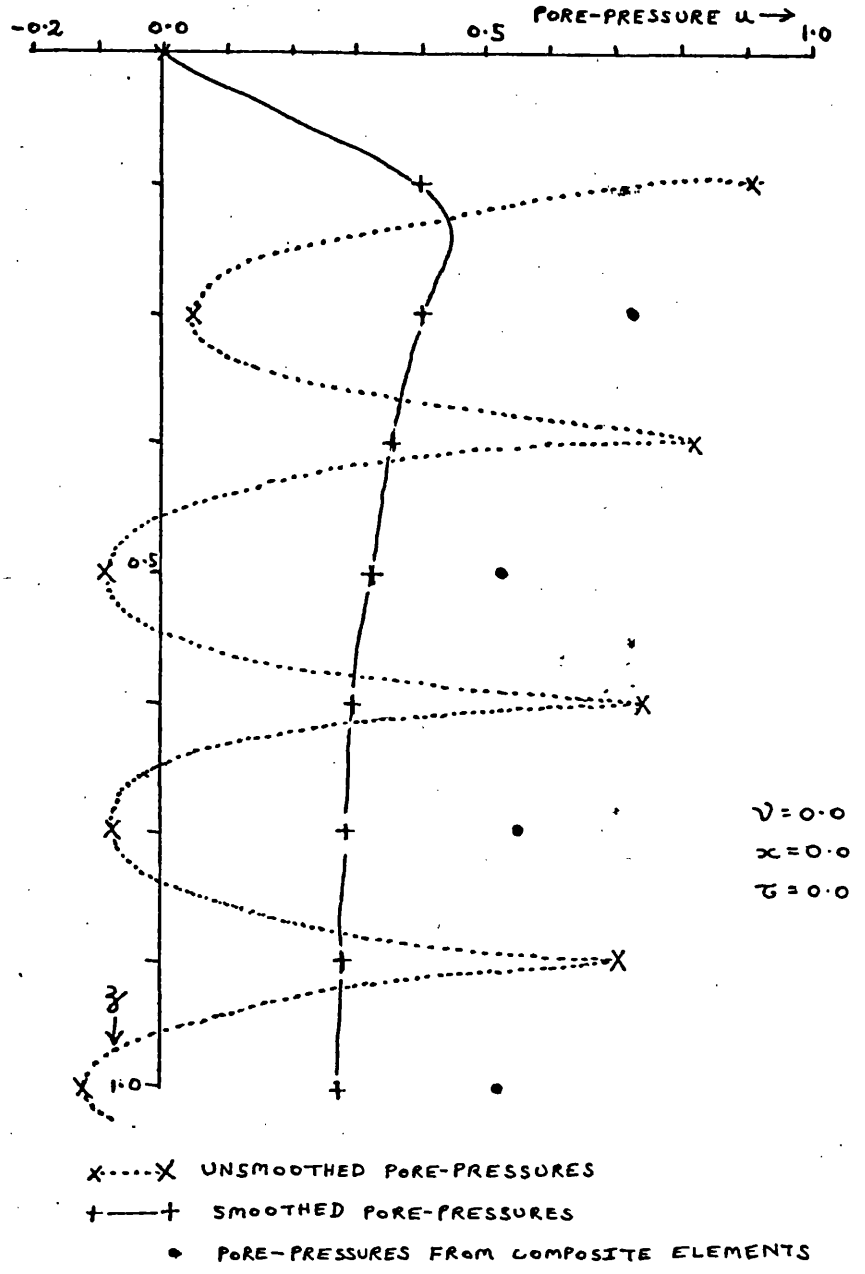


Fig. 6.2.3 FINETIM unsmoothed & smoothed pore-pressures

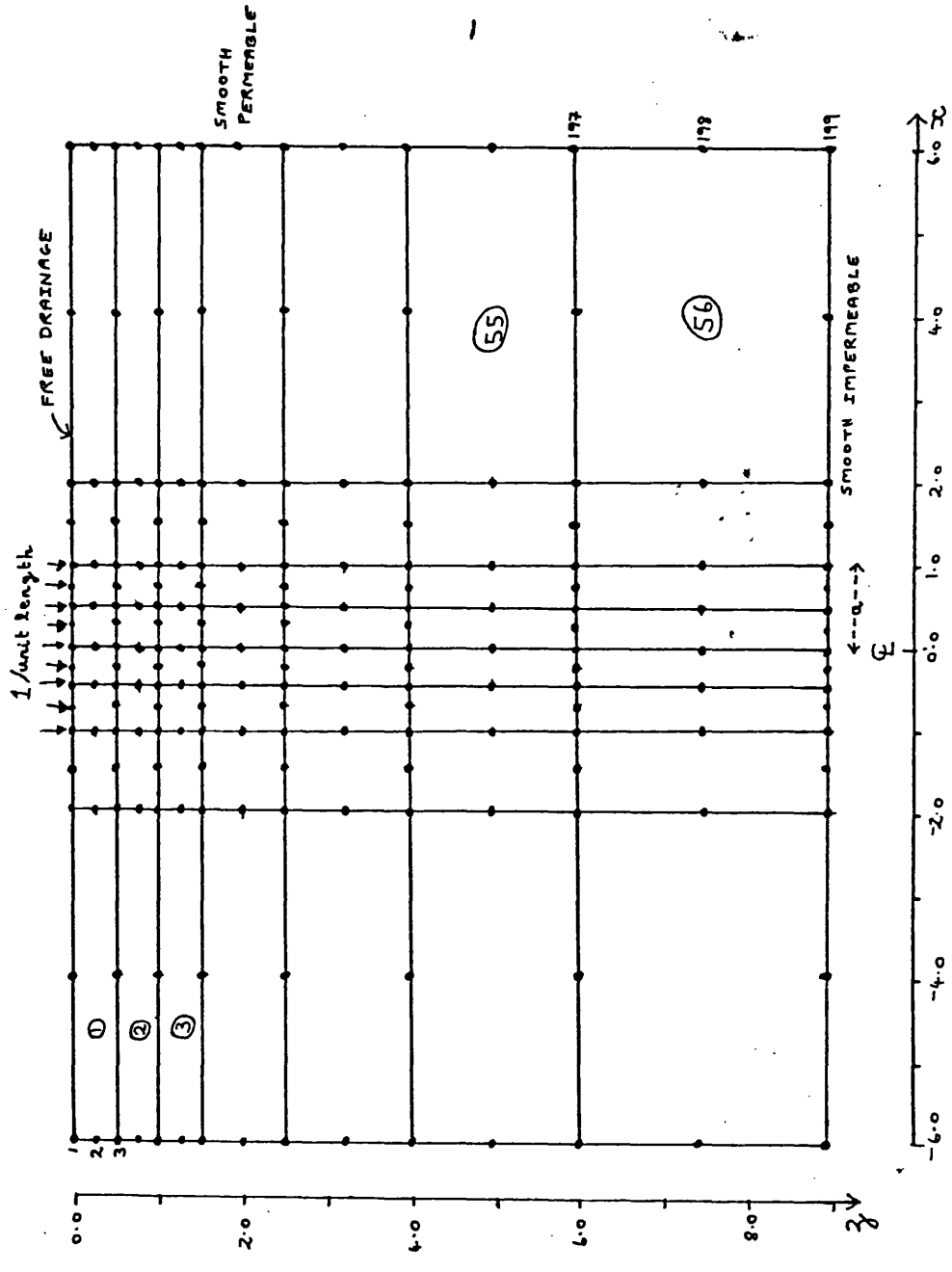


Fig. 6.2.4 Finite element mesh for half-space problem

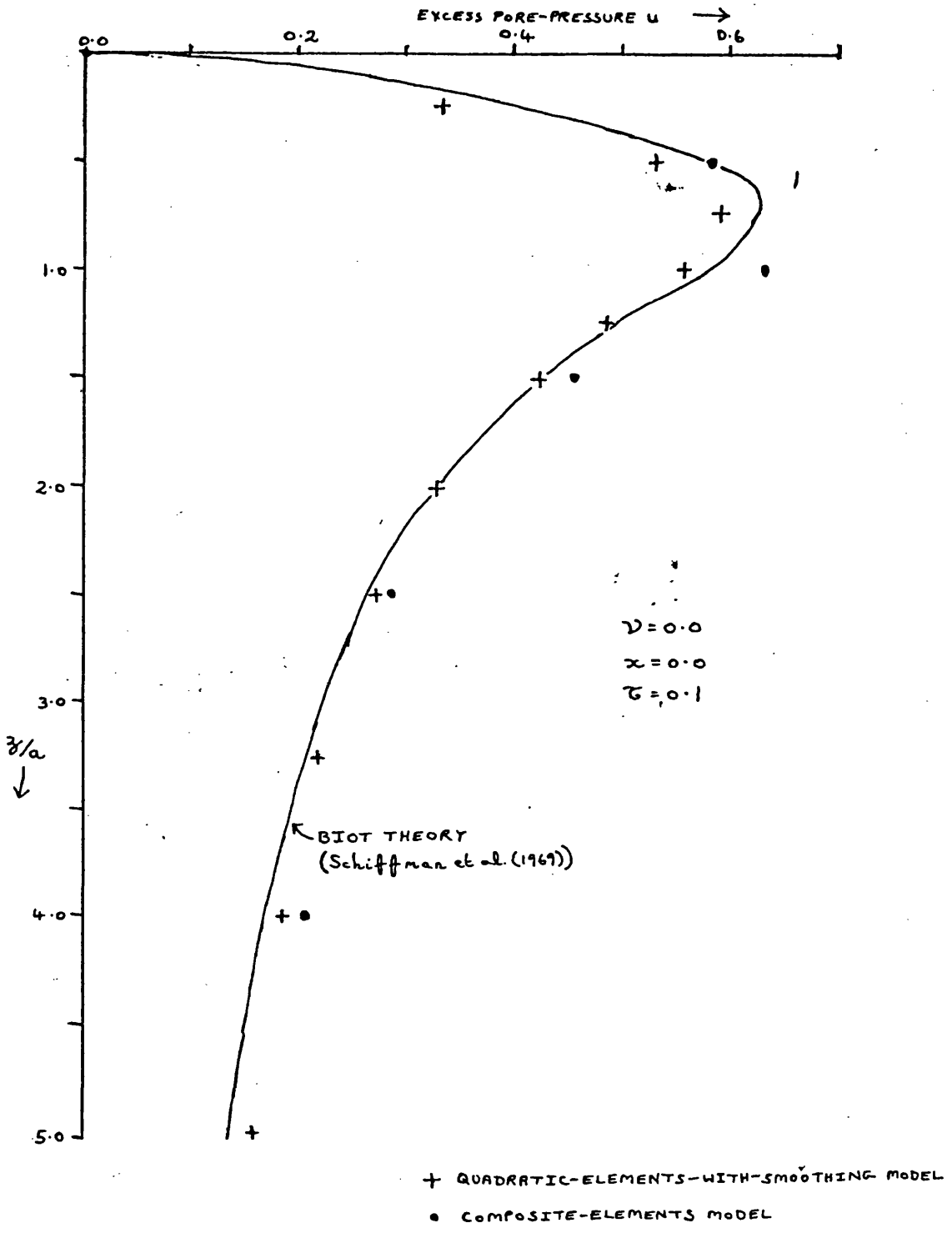


Fig. 6.2.5 FINETIM: pore-pressure distribution (vertical)

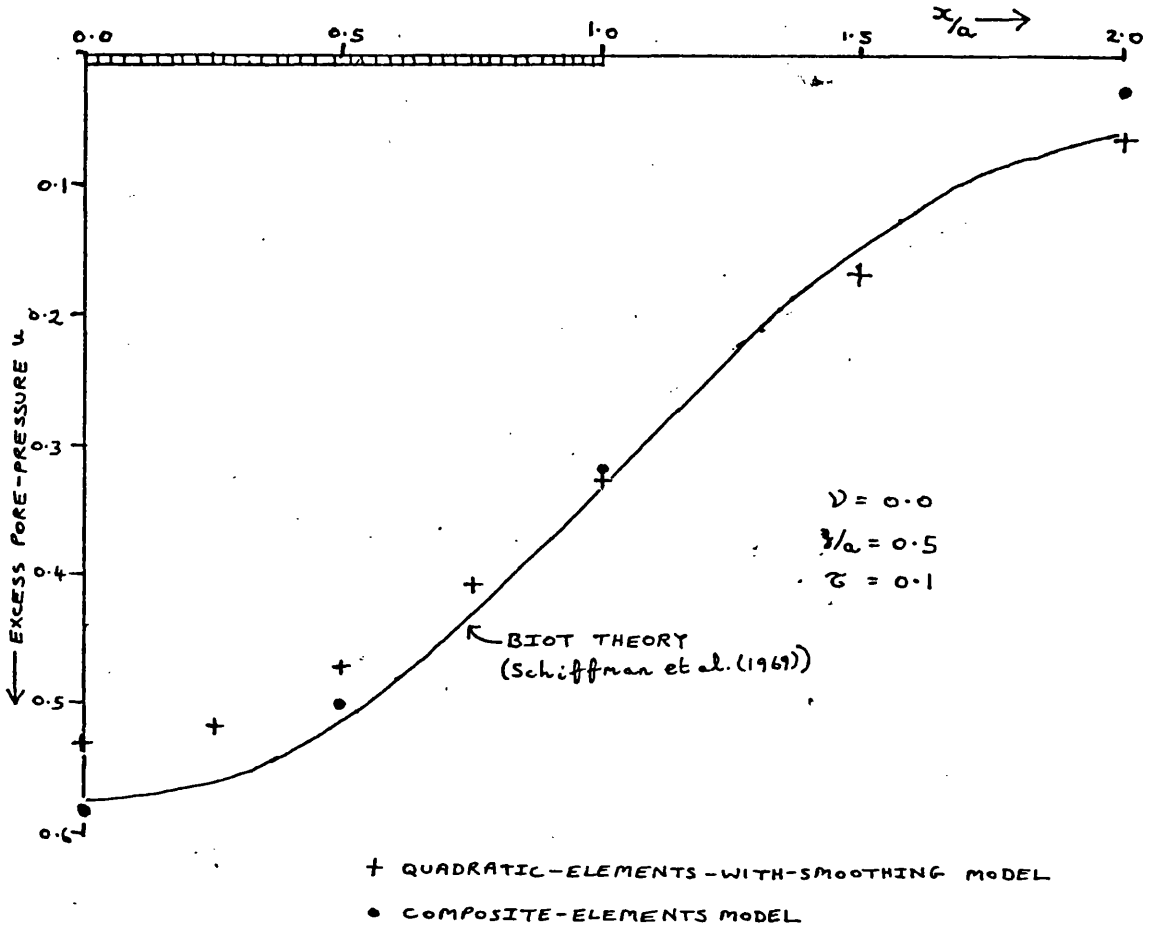


Fig. 6.2.6 FINETIM: pore-pressure distribution (horizontal)

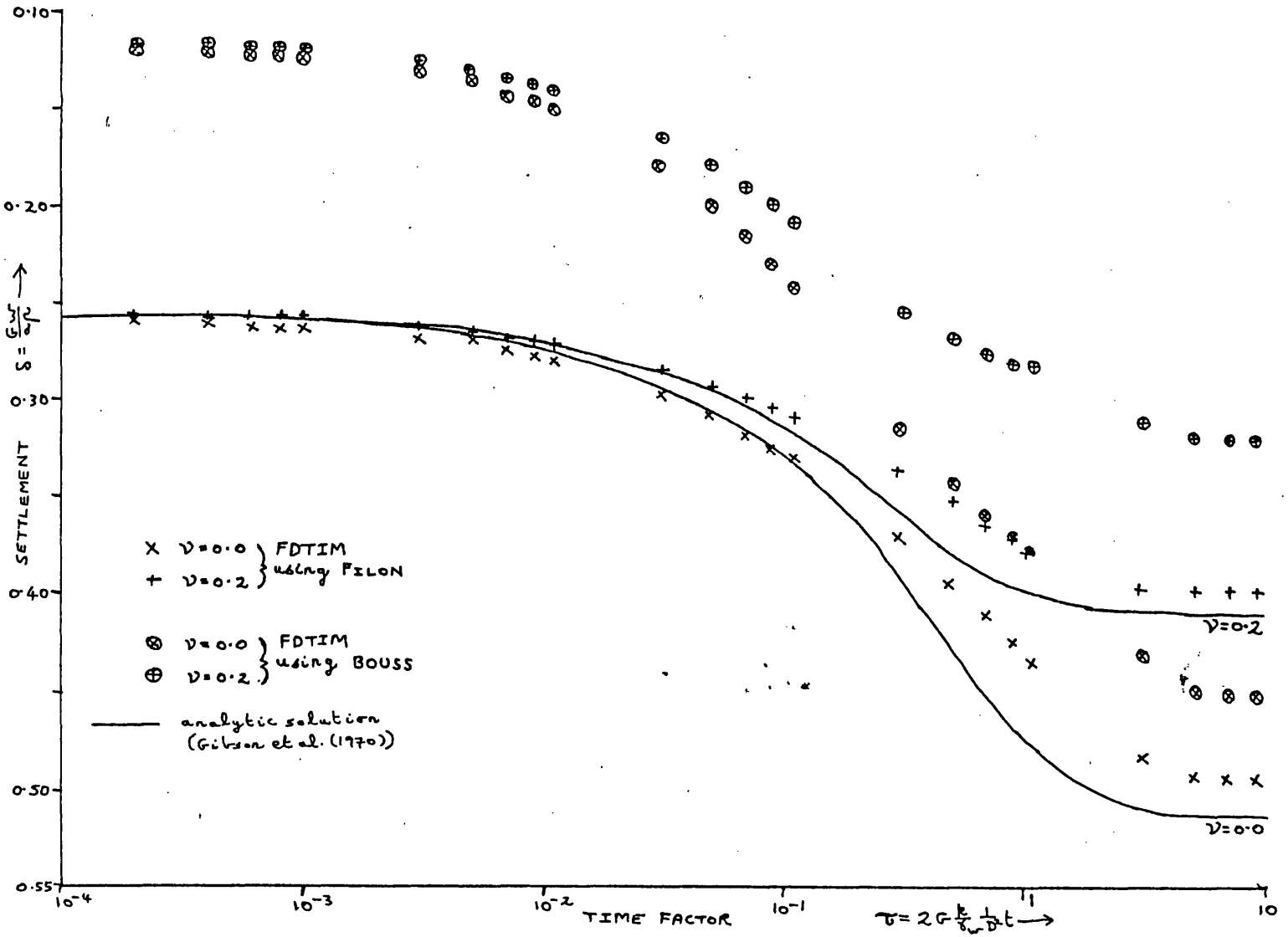


Fig. 6.2.7 FDTIM time - settlement curves

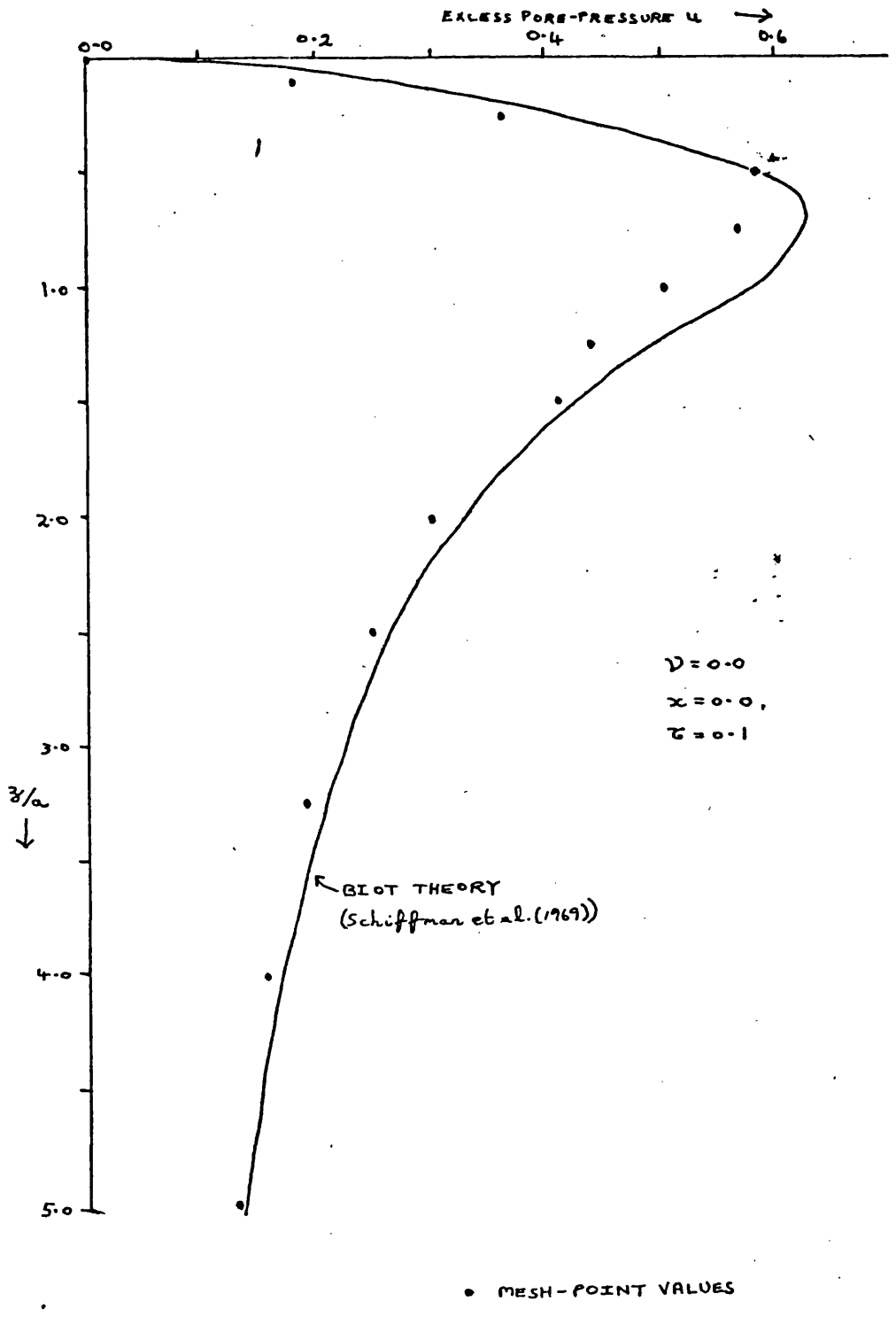


Fig. 6.2.8 FDTIM: pore-pressure distribution (vertical)

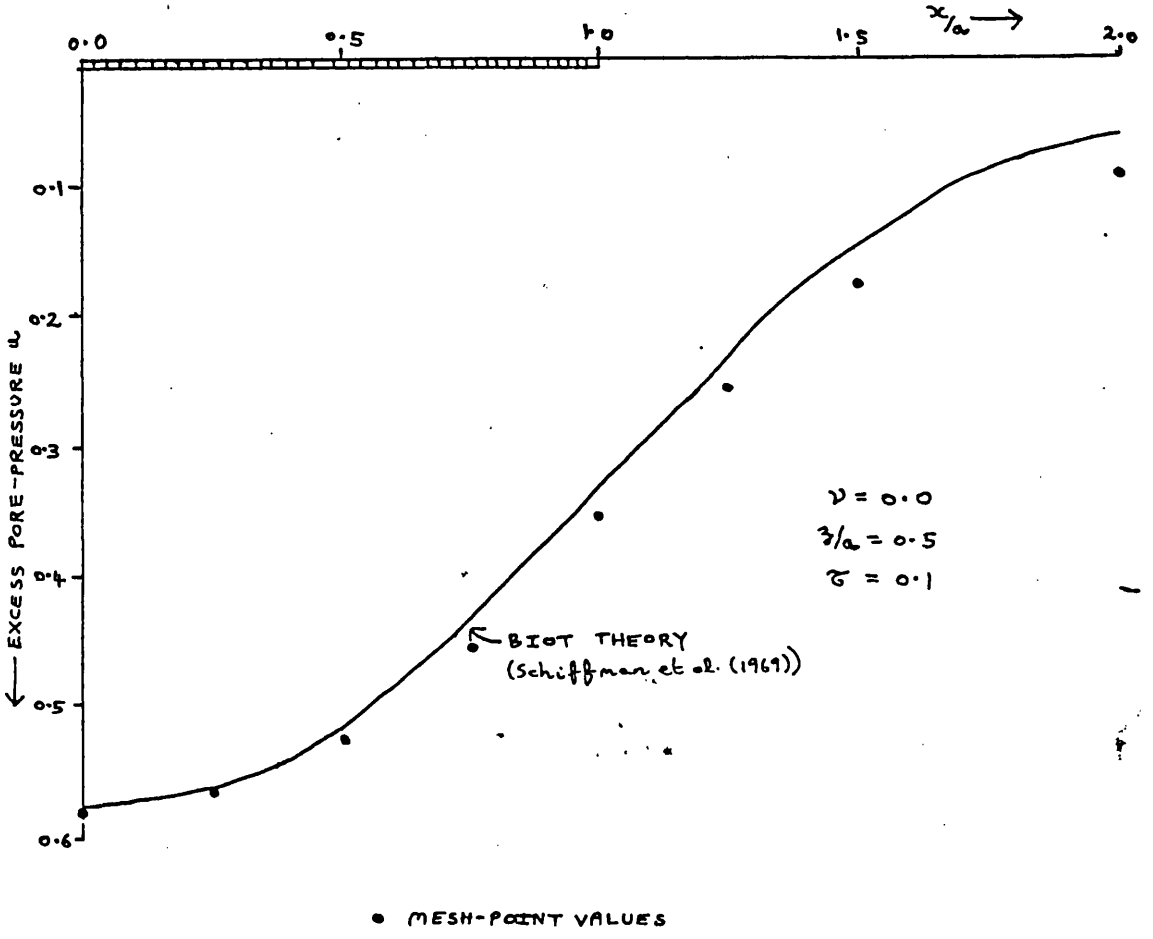


Fig. 6.2.9 FDTIM: pore-pressure distribution (horizontal)

6.3 Rigid footings

An alternative method of testing the validity of numerical results is to perform an instrumented test on a soil of known properties. This has not been attempted (or, at least, not reported) as frequently as comparisons with analytic solutions, although Davis and Poulos successfully verified their theoretical predictions for settlement rates of a flexible footing, by means of a very careful series of model footing tests in a specially-built pressurized brass tank. (The apparatus is described in Davis and Poulos (1968) and the results analysed in Davis and Poulos (1972). Burland (1972) has reported similar tests.)

With rather more modest modelling and testing facilities available, a series of plane strain rigid footing tests was nevertheless conducted in the School of Architecture and Building Engineering. The apparatus is shown in fig 6.3.1 as set up for the final three-footing interaction test. The box was made of 25mm plywood, with a removable front panel of transparent Makrolon. The footings were of 9mm steel, and the beam connecting them of 3mm steel. Loads were applied by deadweight hangers at the points indicated. Porous plates were placed on the soil base and surface. Footing displacements were measured by linear variable differential transducers, connected via an amplifier to pen-recorders. Full details of the tests are given in the project report, Vahabi (1978).

The soil used was a remoulded Devolite clay, which was tested using a triaxial apparatus, to determine its permeability, Young's modulus and Poisson's ratio. Vahabi obtained the following values of the soil parameters for the moisture content of 36% at which the test were conducted:

$C_u = 12.0 KN/m^2$ $\quad E_u = 180.0 KN/m^2$ $\quad M_v = 0.0714 m^2/MN$

$\nu' = 0.498$ $\quad E' = 179.8 KN/m^2$ $\quad k = 0.14 \times 10^{-7} m/sec.$

The three-footing situation will be considered in Chapter 7. Here, we will discuss an earlier test with only one footing, loaded instantaneously with a 10Kg weight (equivalent to a load of $17.09 KN/m^2$ over the footing). Vahabi conducted this test

twice, obtaining the time-settlement curves shown in fig 6.3.4. For the amount of consolidation settlement occurring, the above value of Poisson's ratio is clearly inaccurate, and in the analyses which follow a value of $\nu' = 0.4$ was arbitrarily taken.

The only theoretical complication compared to the problems of the previous section, is the rigidity of the footing. We can therefore use this problem to illustrate the stress distribution theory elaborated in §2.2 and incorporated in the finite difference program FDTIM. Since the problem is symmetric about the centre-line, only half the problem was modelled, using a mesh of 20 columns and 16 rows.

In §2.2 we described two adaptations to the standard Boussinesq stress distribution theory:

(i) the non-uniform contact pressure distribution across a rigid footing, given by a formula (equation 2.2.19) involving a parameter m, $2 \leq m < \infty$, chosen according to the relative depth of the soil layer (see fig 2.2.6);

(ii) the influence on the stress distribution in the soil of the rigid base underlying it, which using the theory of Filon (1903) in §2.2 gives rise to the stress distribution subroutine FILON.

To separate the effects of these two adaptations, we first show in fig 6.3.2 the profiles of immediate settlements obtained using FDTIM with the infinite-depth Boussinesq stress-distribution formulae 2.2.9-11. The program was run for the case of a flexible footing (uniform contact pressure across the footing), giving curve (i) in fig 6.3.2, and then using Borowicka's formula 2.2.16 for the contact pressure across a rigid footing on a half-space. This formula is equivalent to setting $m = 2$ in equation 2.2.19, and the resulting settlement profile is shown in curve (ii). We see that the Borowicka formula results in a much more even settlement across the footing, though a slight hogging occurs, with the footing edge settling about 7% more than the centre. This may be compensated for by adjusting the parameter m. Since the footing half-width is 19mm

and the soil depth is 75mm, we may consult fig 2.2.6 for m_{opt} with $\nu = 0.5$ at a relative depth of 4, and obtain $m_{opt} = 2.12$. Use of this value gives the settlement profile shown in curve (iii) in fig 6.3.2, with the hogging reduced to a little more than 1%. Further trial-and-error adjustment of m could be made to eliminate the hogging altogether. It is interesting to note that the traditional approximation for a rigid footing settlement as the average of the centre and edge settlements for a flexible footing, would in this case underestimate the correct value by only 3-4%.

With the use of the parameter m demonstrated, we introduce the finite layer stress distribution theory as embodied in the subroutine FILON. In this case there is no need to increase the parameter m above 2.0 - see fig 2.2.6 - and we show the immediate settlement profiles for both flexible and rigid footings in fig 6.3.3. As compared to the Boussinesq-theory results presented above, the centreline settlements are increased by about 50% for both flexible and rigid footings.

We now turn to the time-dependent deformation of the model, and compare it with the footing settlements as observed by Vahabi. For this purpose we use the FILON stress subroutine with $m = 2.0$, and obtain the time-settlement graph in fig 6.3.4. For comparison, a finite element analysis of the problem was also made using the program FINETIM and the three-footing model described in Chapter 7, with a very low Young's modulus given to the bar linking the footings so as to avoid any interaction between them. It is seen that while the time-settlement predictions from FDTIM and FINETIM are in tolerable agreement, they are 100-200% in excess of the observed settlements.

Many reasons may be proposed to account for this discrepancy. The more important may be summarized as follows:

(i) Young's modulus values obtained from laboratory tests are frequently found to underestimate in situ values by up to one order of magnitude;

(ii) the sides of the experiment box were not completely rigid, but bulged outwards during the initial consolidation of the soil,

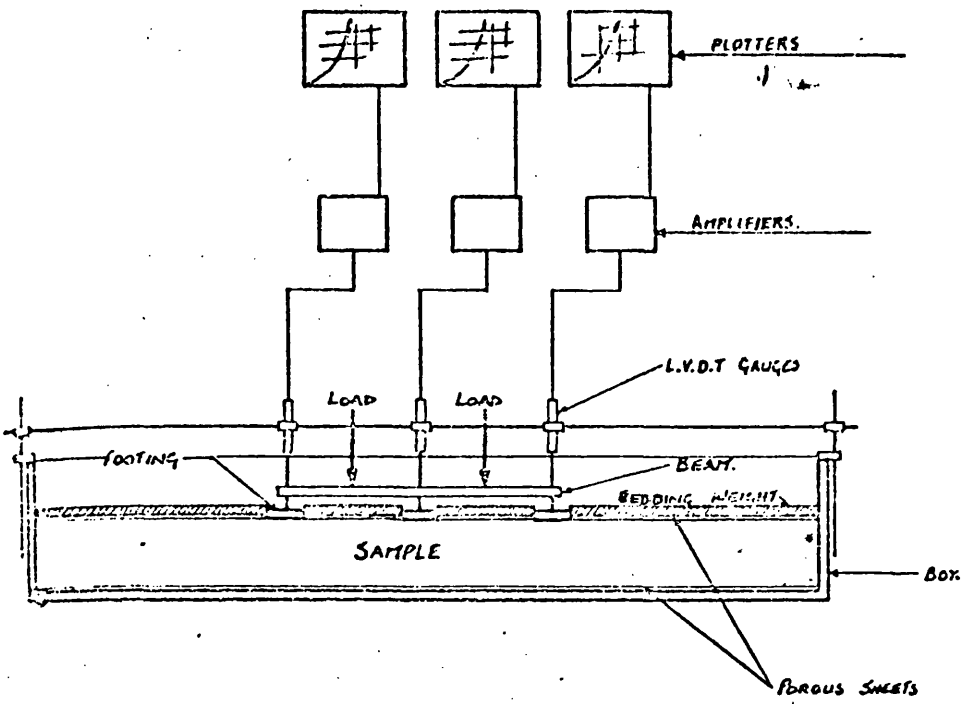
destroying the plane-strain condition for the subsequent tests;

(iii) heavy steel plates were applied as bedding weights to the soil surface during the tests, to prevent heaving away from the footing;

(iv) the small scale of the test causes edge effects to be significant.

Each of these reasons could cause the observed settlements to be smaller than those predicted. In the absence of sizeable errors caused by experimental defects, such as (ii) and (iii) above, it would be valid to adjust the values of E and ν used in the programs until agreement with observed results was reached; such a method of determination of in situ soil parameters by back-analysis from tests is often used. However, it is felt that the probable experimental errors are too large to warrant this here.

One or two encouraging points do emerge from the results in fig 6.3.4. There is good agreement between the finite element and finite difference results throughout the consolidation process, indicating the validity of the rigid-footing theory used in FDTIM. Also, the predicted rates of consolidation correspond to those observed, indicating that the measured soil permeability is accurate. To test the importance in this problem of using a two-dimensional consolidation model, rather than the simple one-dimensional consolidation theory, the programs were also run with the horizontal permeability k_x set to zero. Very similar settlement rates were obtained, suggesting that nearly all the flow of pore-water occurs vertically.



(Vahabi (1978))

DIMENSIONS

LENGTH	900 mm
SOIL DEPTH	75 mm
FOOTING DIMENSIONS	38 mm x 151 mm x 9 mm
DISTANCE BETWEEN FOOTING MID-POINTS	150 mm
OUT-OF-PLANE THICKNESS	151 mm
BEAM DIMENSIONS	320 mm x 38 mm x 3 mm

Fig. 6.3.1 Model footing test apparatus

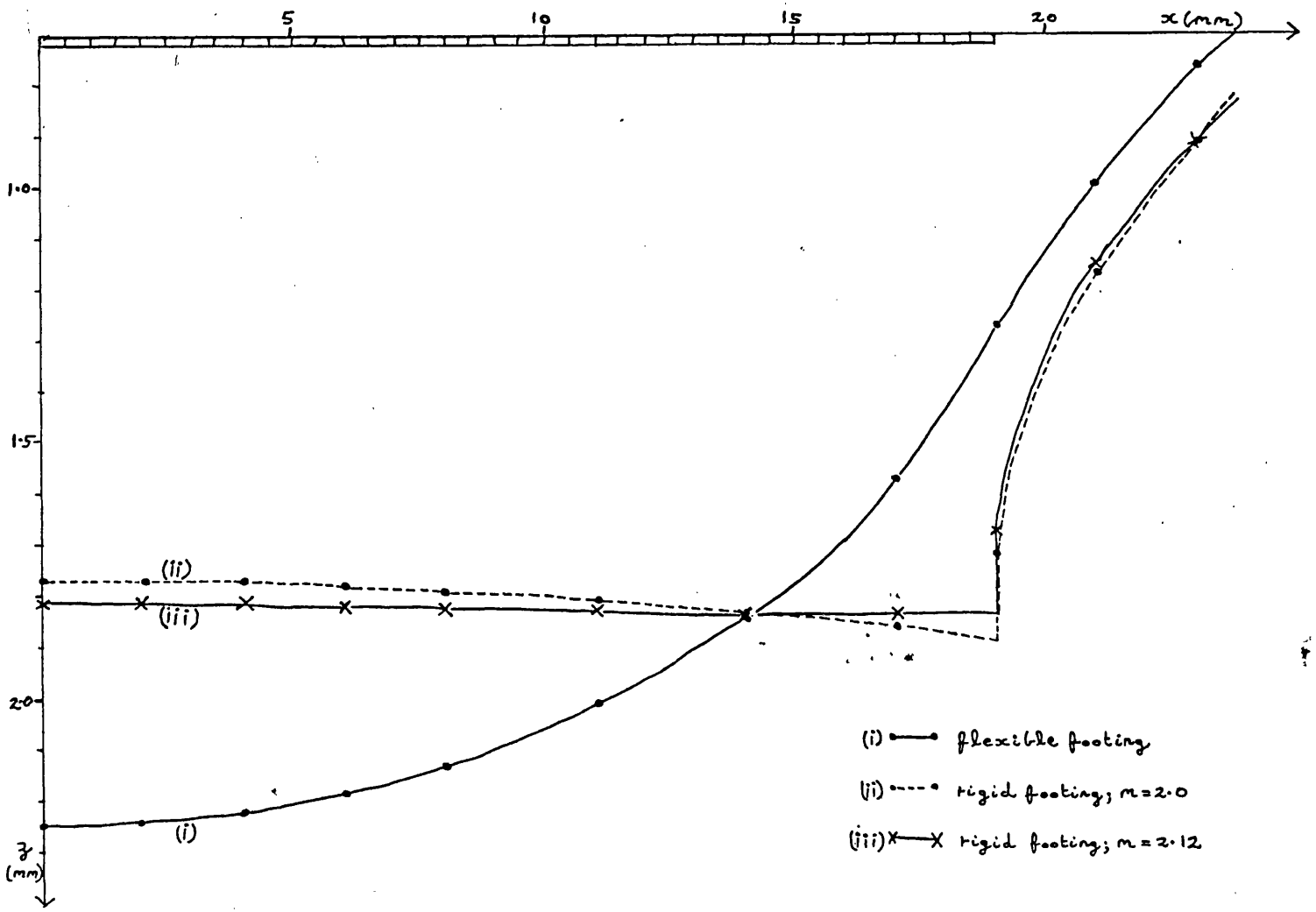


Fig. 6.3.2 Settlement profiles; Boussinesq stresses

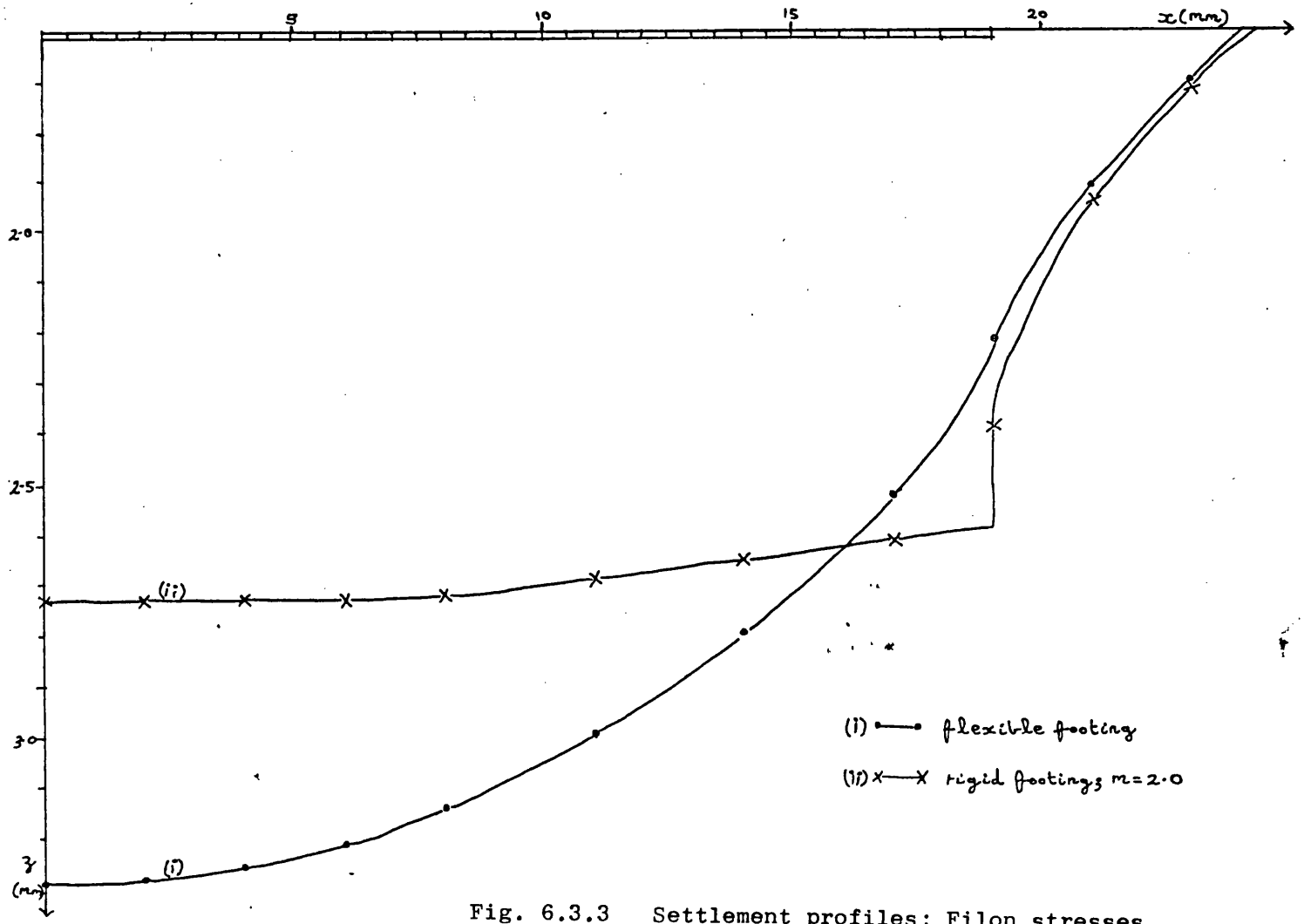


Fig. 6.3.3 Settlement profiles; Filon stresses

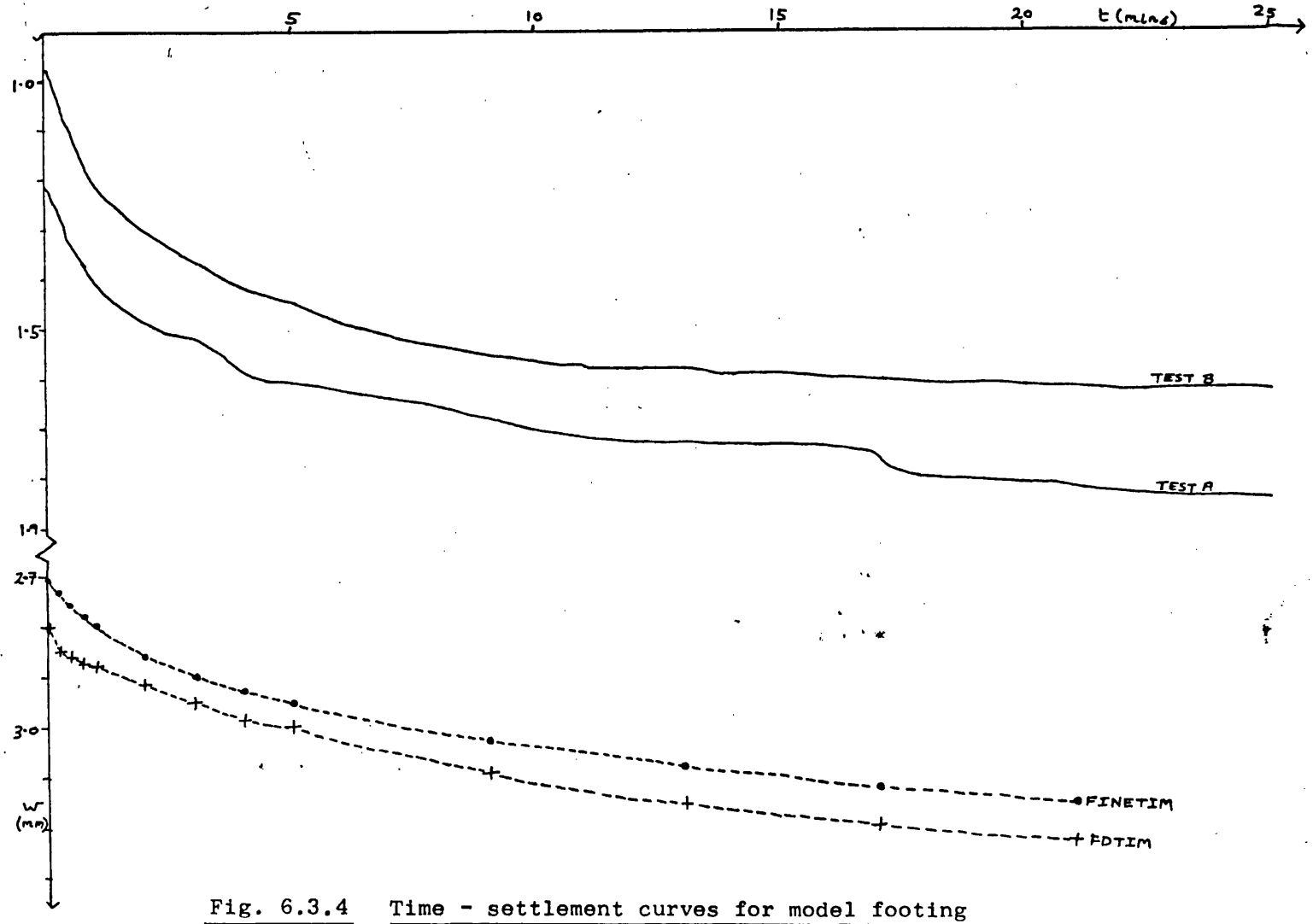


Fig. 6.3.4 Time - settlement curves for model footing

6.4 Conclusions

6.4.1 It has been shown in §6.2 that for two sample plane strain consolidation problems involving flexible footings the computer programs FINETIM and FDTIM give results in good agreement with analytic solutions. In particular, the finite difference program FDTIM with the FILON stress distribution subroutine developed in §2.2 performs equally as well as the finite element program with a saving of around 400% on computer run-time. In FINETIM, the pore-pressure smoothing process developed in §4.3 appears to give a small increase in accuracy of initial displacements and pore-pressures, provides nearly twice as many nodal pore-pressure values and allows a mesh of standard quadratic isoparametric elements to be used, all this at the cost of about 20% more computer run-time compared to the composite-element mesh. In one-dimensional analyses, finite difference analyses take about half as long as finite element ones (Desai (1975)).

6.4.2 If we accept that the experimental results for the rigid footing problem of §6.3 are invalid for the reasons given and that the finite element solution has the same accuracy as in the problems of §6.2, then the finite difference program FDTIM again performs satisfactorily. In particular, the method of modelling rigid footings, as developed in §2.2, gives sufficiently realistic footing settlement profiles and accurate centreline settlements.

6.4.3 The Boussinesq theory for stress distribution through a half-space is found to be invalid for finite soil layers, giving an error in surface settlements of around 50% for a footing on a soil layer of relative depth (depth to footing half-width) of 4. Several recent finite difference programs, for example Wood (1972) and Murray (1974), use infinite-depth stress distribution theories, and the results presented here would suggest that this can cause significant underestimation of settlements even for relatively deep layers. At the other extreme, Davis and Poulos' (1972) program uses the cumbersome formulae of Pickett (1938) requiring numerical integration, etc. It has been found that the FILON finite-layer subroutine used here has a very modest extra cost over the Boussinesq theory (an extra 2½ seconds for 275 mesh-points) while providing much more accurate settlement predictions.

CHAPTER 7 Results (ii): embankments and frame problems

7.1 Introduction

Having considered in chapter 6 the performance of the programs FDTIM and FINETIM on homogeneous-soil single-footing problems, we here turn to more complex situations. In §7.2 we model the consolidation of a trial embankment which was constructed on compressible alluvium. The main complexity here is the layered nature of the soil, with abrupt changes in the soil parameters down a vertical cross-section of the soil layer.

Then in §7.3 we use the computer programs to model the settlement of three footings linked by a steel beam loaded at the mid-points of the two bays. In contrast to the previous chapter, it is found that the introduction of these complexities severely reduces the efficiency of the finite difference program FDTIM, so that a finite element analysis becomes the best method of solution.

7.2 Embankments

The input data used to study the embankment type of loading were derived from a full-scale trial embankment constructed in 1965 on compressible alluvium at Avonmouth as part of the site investigations for the M5 motorway. Details of the construction and of the measurements of settlement and pore-pressure obtained subsequently are contained in a Road Research Laboratory report by Murray (1971). From this report fig 7.2.1 (a) and (b) are taken, showing a plan of the site and a section through the soil under the embankment. The soil stratification may be simplified to a four-layer system overlying Keuper marl bedrock, with the soil parameters given in fig 7.2.2. The values of compressibility were determined from laboratory consolidation tests, and are usually assumed to be correct to within ±20%. The Young's modulus values have been deduced from these, assuming a drained Poisson's ratio of 0.28 as typical for normally consolidated clays. Coefficients of permeability were measured in two ways:

(i) laboratory tests on undisturbed samples, which should in theory give the vertical permeability k_z;

(ii) in situ permeability tests carried out following the method proposed by Gibson (1963). These give a measurement of the all-round permeability $k_m = \sqrt{k_x k_z}$.

From these measurements of k_z and k_m we may deduce the horizontal permeability k_x. Values of k_m, k_z and k_x for each soil layer appear in fig 7.2.3, and it is seen that the horizontal permeability exceeds the vertical permeability quite considerably, the ratio Q between the two ranging from 42 in the top layer to 224 in the bottom. In a later trial embankment analysis Murray and Symons (1974) used a finite difference consolidation program to obtain time-settlement curves for the three cases:

(i) laboratory test result for both vertical and horizontal permeability

(ii) laboratory and in situ test results to deduce k_x and k_z as above

(iii) in situ test result for both vertical and horizontal permeability.

He found that (i) gave the slowest, and (iii) the most rapid consolidation, and that (iii) corresponded very well to the observed settlement rate. This would indicate that the laboratory determination of permeability is inaccurate, and that in reality the soil permeability in each direction is close to the value obtained from field tests. Unless otherwise stated, case (iii) has been used in the numerical models below.

From piezometer readings taken before construction started, Murray concluded there was a porous drainage layer at the soil base, and in his finite difference analysis he took a value of 1.0 for the Skempton pore-pressure coefficient A.

We shall briefly consider the effects of these and other conditions on the centreline settlement with time, for the simplest case of a linear elastic soil model and instantaneous application of the embankment load at $t = 0$.

In creating a mesh for the analysis of the problem by the finite difference program FDTIM, severe stability problems were encountered with all but the coarsest mesh in the z-dimension, even with the minimum possible timestep of $\Delta t = 1$. Fig 7.2.4 shows the initial centreline pore-pressure profile, and that after the first timestep, for the meshes:

(a) $\Delta z = 0.25$ throughout mesh

(b) $\Delta z = 0.5$ throughout mesh

(c) $\Delta z = 0.5$ up to $z = 5.0$, 1.0 up to $z = 9.0$, 1.5 up to $z = 12.0$.

Only the last mesh gives a reasonable result. The errors in the other results seem to occur mainly around the internal boundaries between soil layers, and approaching the base. (This problem was mentioned in §3.3.) We present below time-settlement curves obtained using mesh (c), although because of the coarseness of the mesh the results are not accurate. (The immediate centreline settlement from mesh (c) was 15% less than that predicted by the finer mesh (a), for example.) They may, however, be used to observe the qualitative effect of varying boundary conditions or parameters.

A finite element mesh of 96 composite rectangular elements was used to model the problem with FINETIM, as shown in fig 7.2.5.

Variation of pore-pressure coefficient A

From fig 7.2.6 it is seen that there is a significant difference in the immediate settlements as predicted by the finite element model (130mm) and the finite difference model (230mm) using the ideal elastic value of $^1/_3$ for Skempton's pore-pressure coefficient A. This difference can not be explained by the coarseness of the finite difference mesh, since the immediate settlement from the finer mesh (a) above was even greater (270mm). As consolidation progresses, however, both solutions approach similar drained settlements of 375-400mm. If the value of A is increased to 1.0 (as is appropriate for a normally consolidated clay - see fig 2.4.1) the undrained excess pore-pressures are increased and the resulting time-settlement curve is much closer to the finite element solution. Such a difference was not noticed in the problems of chapter 6; the writer would suggest that the extreme relative shallowness of the soil layer in this case emphasizes the effect on the pore-pressures of the redistribution of stresses in the soil when settlement occurs. This effect is taken into account in the Biot consolidation theory used in the finite element program, but not in the Terzaghi diffusion-type theory (see §3.2). The effect may be introduced into the latter theory by increasing A above the theoretically correct value of $^1/_3$. We have therefore used A = 1.0 in the results presented below.

Rough/smooth base

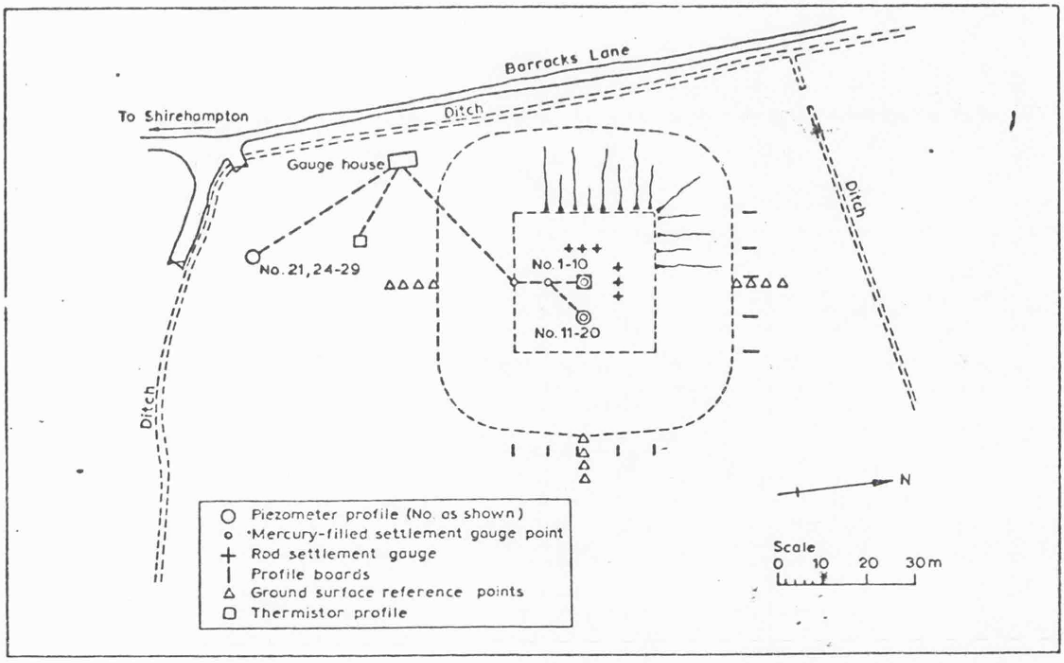
The curves in fig 7.2.6 were obtained on the assumption of a smooth impermeable base. For the finite difference model the finite-layer FILON stress subroutine was used. Fig 7.2.7 shows the effect of modelling a perfectly rough base in the finite element case, and also of using the infinite-depth Boussinesq stress formulae in the finite difference case. The correct stress distribution should take into account both the rigidity and the roughness of the base, although in effect the Boussinesq theory ignores both. Fig 7.2.7 shows that the two complicating factors largely cancel each other out for shallow

soil layers, with the Boussinesq-theory FDTIM curve being very close to that for the rough-base FINETIM solution in the early stages of consolidation. This would perhaps explain the success of programs using Boussinesq theory in predicting short-term settlements for large-scale practical problems.

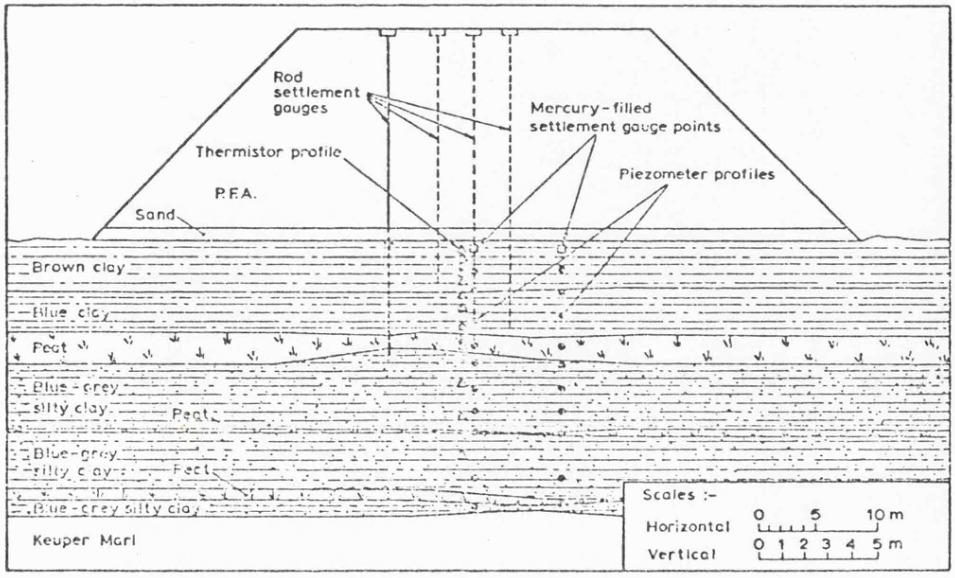
Permeability conditions

The influence of the base permeability on centreline consolidation is seen in fig 7.2.8 using both FDTIM and FINETIM. In both cases, an impervious base slows down the consolidation process. The permeable-base FINETIM curve takes just over a year to achieve 90% consolidation, which corresponds well with the recorded settlement curve. Against this, one might point out that as the embankment is not a true plane strain problem (see fig 7.2.1), the recorded centreline settlement could be expected to be more rapid than predicted. To test the importance of horizontal permeability, the permeable-base FINETIM model was rerun with a horizontal permeability of zero; this made less than $^1/_4$% difference to the settlements, showing that virtually all the water flow was in the vertical direction.

So far, we have used the all-round permeability k_m in both directions; we may ask how the behaviour changes if we use the cross-anisotropic permeabilities k_z and $k_x = Qk_z$ deduced from laboratory and field tests as explained above. The results of running the models using these permeabilities are shown in fig 7.2.9. For both FINETIM and FDTIM, the base condition (permeable or impermeable) makes virtually no difference to the centreline consolidation. Using FDTIM the consolidation is considerably slower than those for isotropic permeability using the field value k_m; this was also observed in a later two-dimensional finite difference analysis by Murray (1974) who concluded that the laboratory value of k_z was indeed inaccurate. Using FINETIM, however, there is little difference between the cross-anisotropy result and that for isotropic permeability k_m with an impermeable base.



PLAN OF SITE SHOWING POSITIONS OF INSTRUMENTS



SECTION ACROSS EMBANKMENT ALONG EAST-WEST CENTRE-LINE

Fig. 7.2.1 Avonmouth trial embankment (from Murray (1971))

Layer	Description	Depth (m)	Compressibility m_v (m^2/KN)	P. Ratio ν' (assumed)	Y. Modulus E' (KN/m^2)	Initial stress state σ_z' (KN/m^2)	Cohesion c' (KN/m^2)	Angle of Friction ϕ'
1.	Brown clay	0-2 m	0.57×10^{-3}	0.28	1.975×10^{3}	16.9	7.18	27°
2.	Blue clay	2-4m	0.32×10^{-3}	0.28	3.519×10^{3}	33.1	9.59	24°
3.	Peat	4-5m	1.50×10^{-3}	0.28	0.750×10^{3}	39.9	0.0	50°
4.	Silty clay	5-12m	0.12×10^{-3}	0.28	9.380×10^{3}	66.1	0.0	33°

Fig. 7.2.2 Soil layers and properties

Layer	Description	All-round permeability (from in situ test) k_m (m/day)	Vertical permeability (from lab. test) k_z (m/day)	Horizontal permeability (from $k_x = k_m^2/k_z$) k_x (m/day)	Ratio $Q = k_x/k_z$
1.	Brown clay	1.728×10^{-4}	2.678×10^{-5}	1.115×10^{-3}	41.6
2.	Blue clay	1.210×10^{-3}	1.412×10^{-4}	1.037×10^{-2}	73.5
3.	Peat	1.556×10^{-4}	1.197×10^{-5}	2.023×10^{-3}	169.0
4.	Silty clay	1.123×10^{-3}	2.371×10^{-5}	5.319×10^{-2}	224.4

Fig. 7.2.3 Soil permeabilities

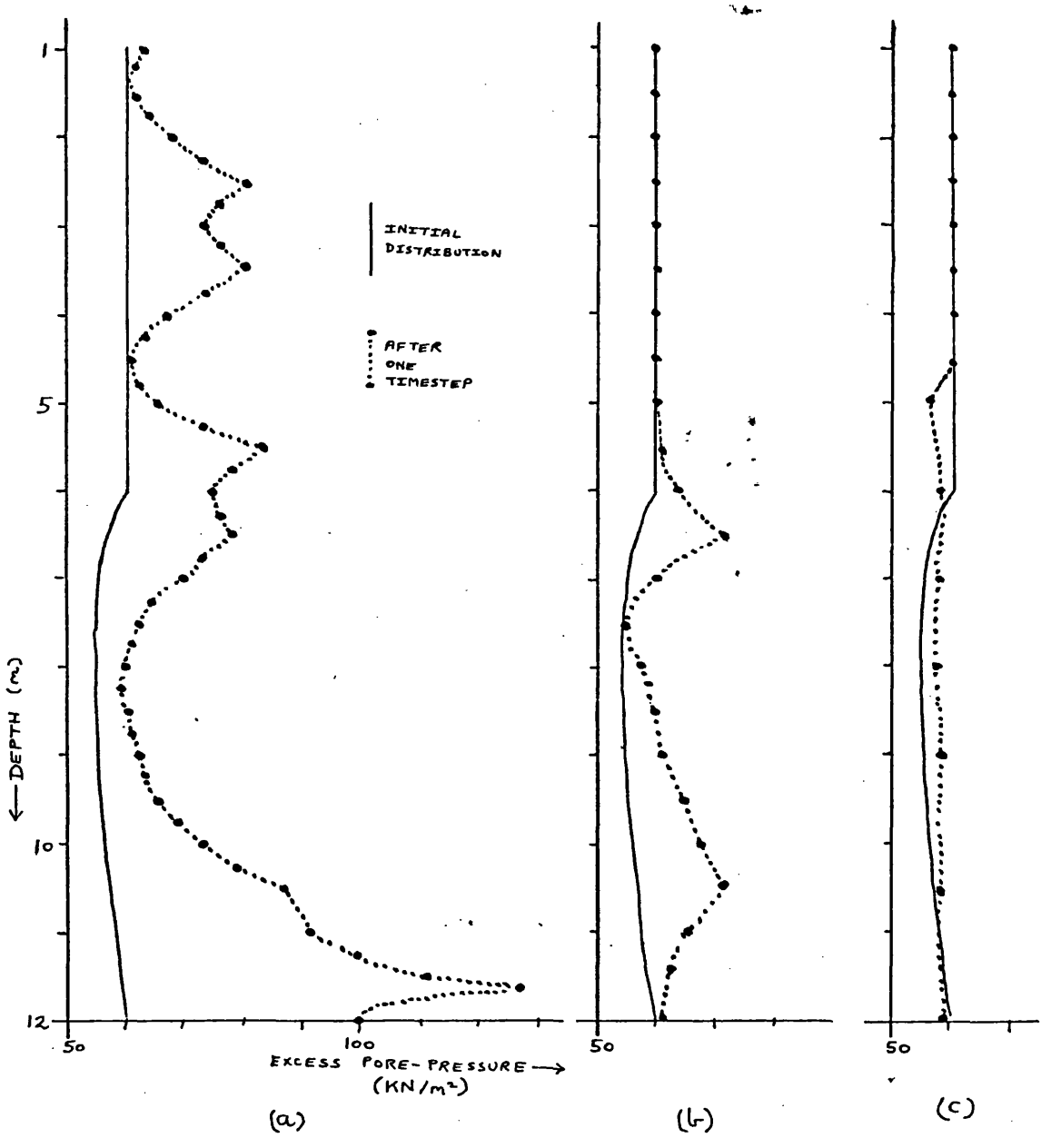


Fig. 7.2.4 Excess pore-pressures for various meshes using FDTIM

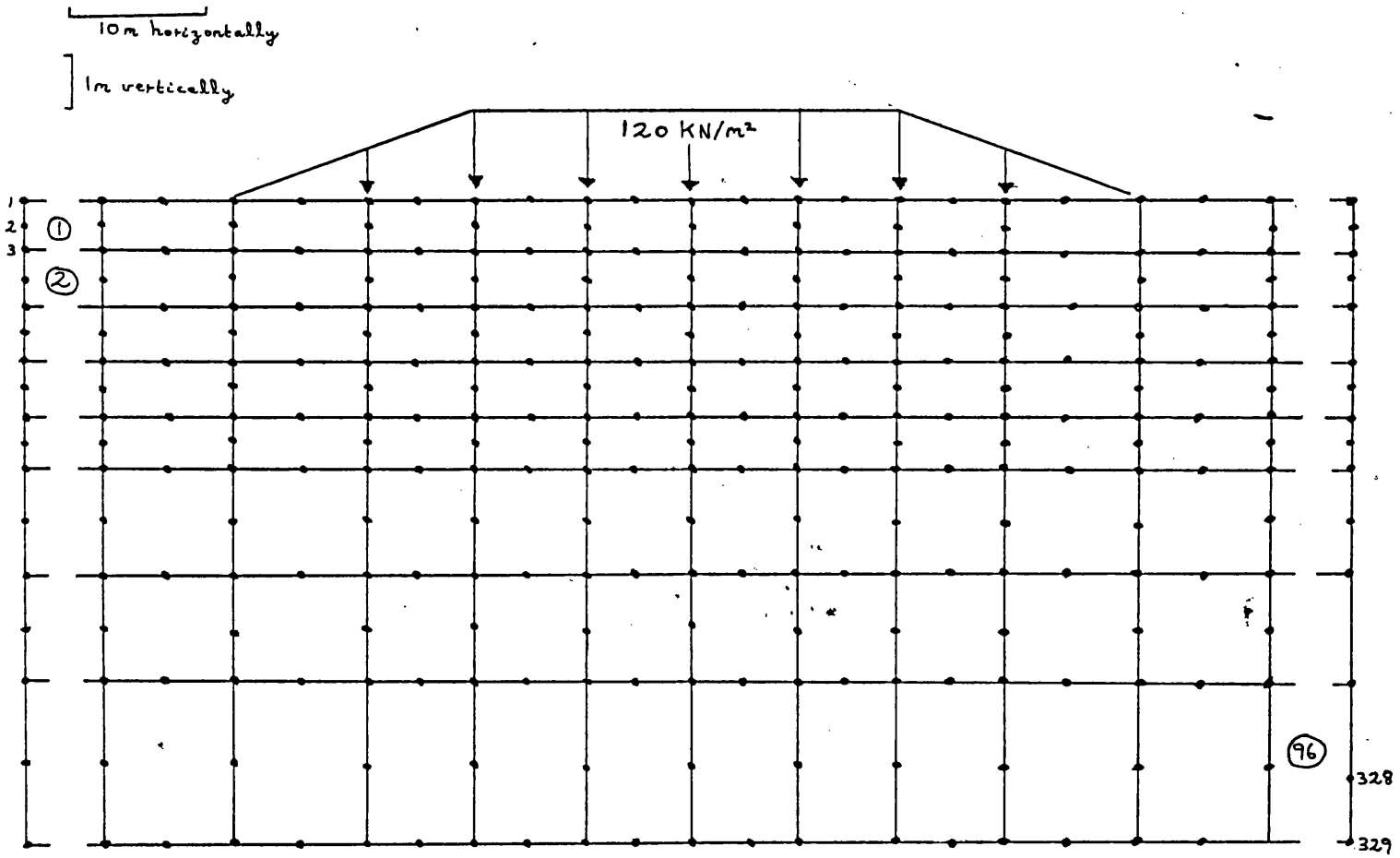


Fig. 7.2.5 Finite element model of embankment

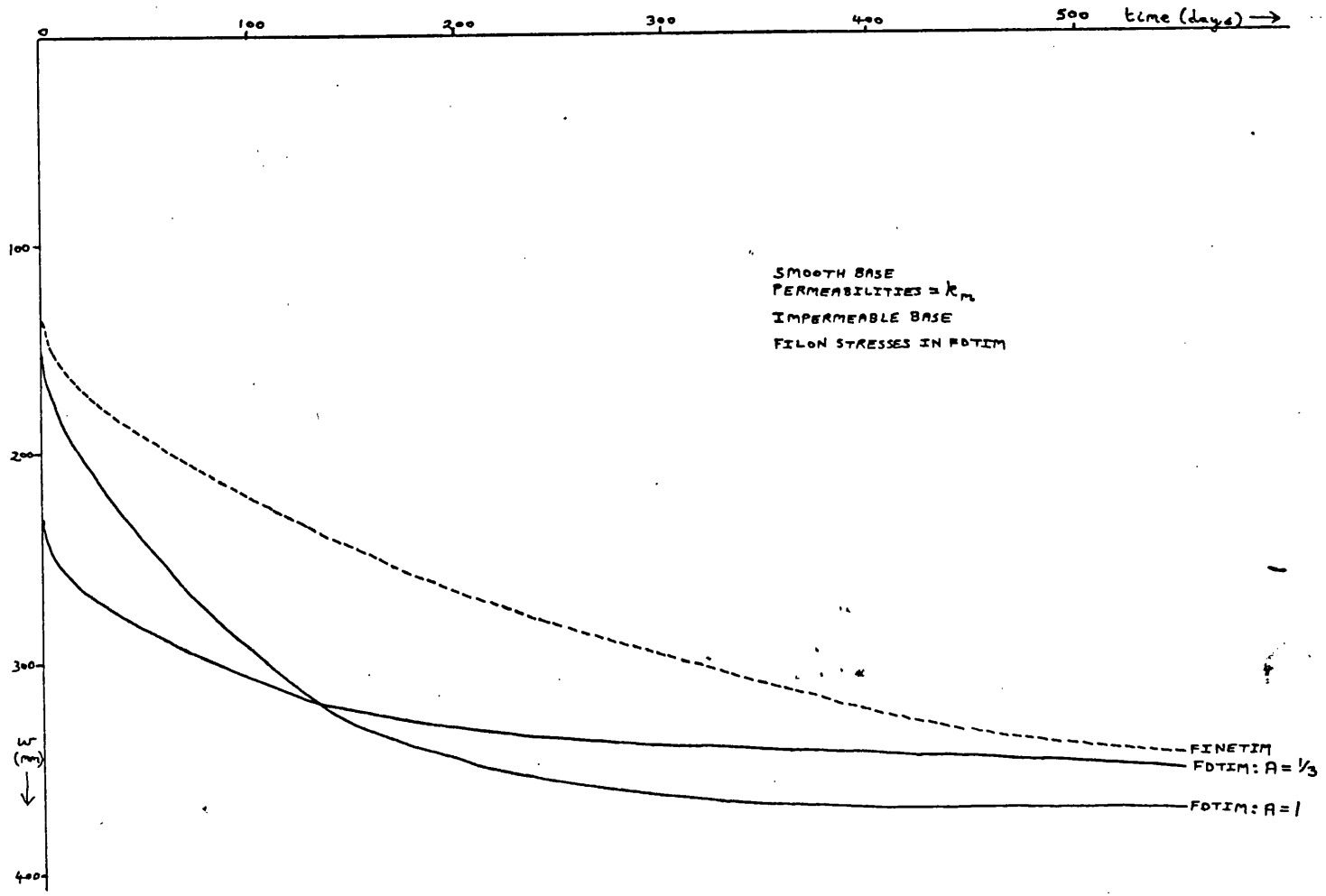


Fig. 7.2.6 Effect of pore-pressure coefficient A on consolidation

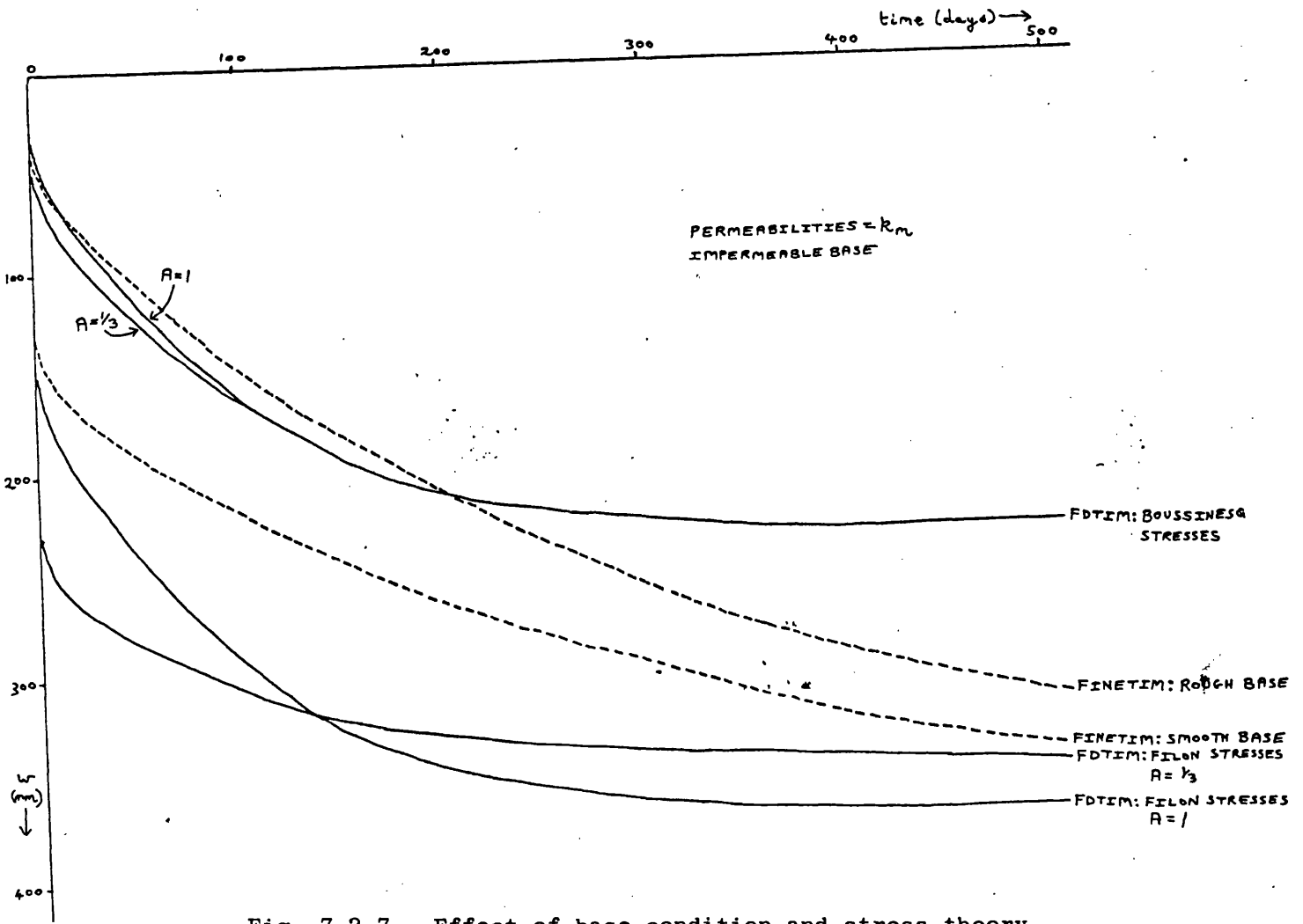


Fig. 7.2.7 Effect of base condition and stress theory

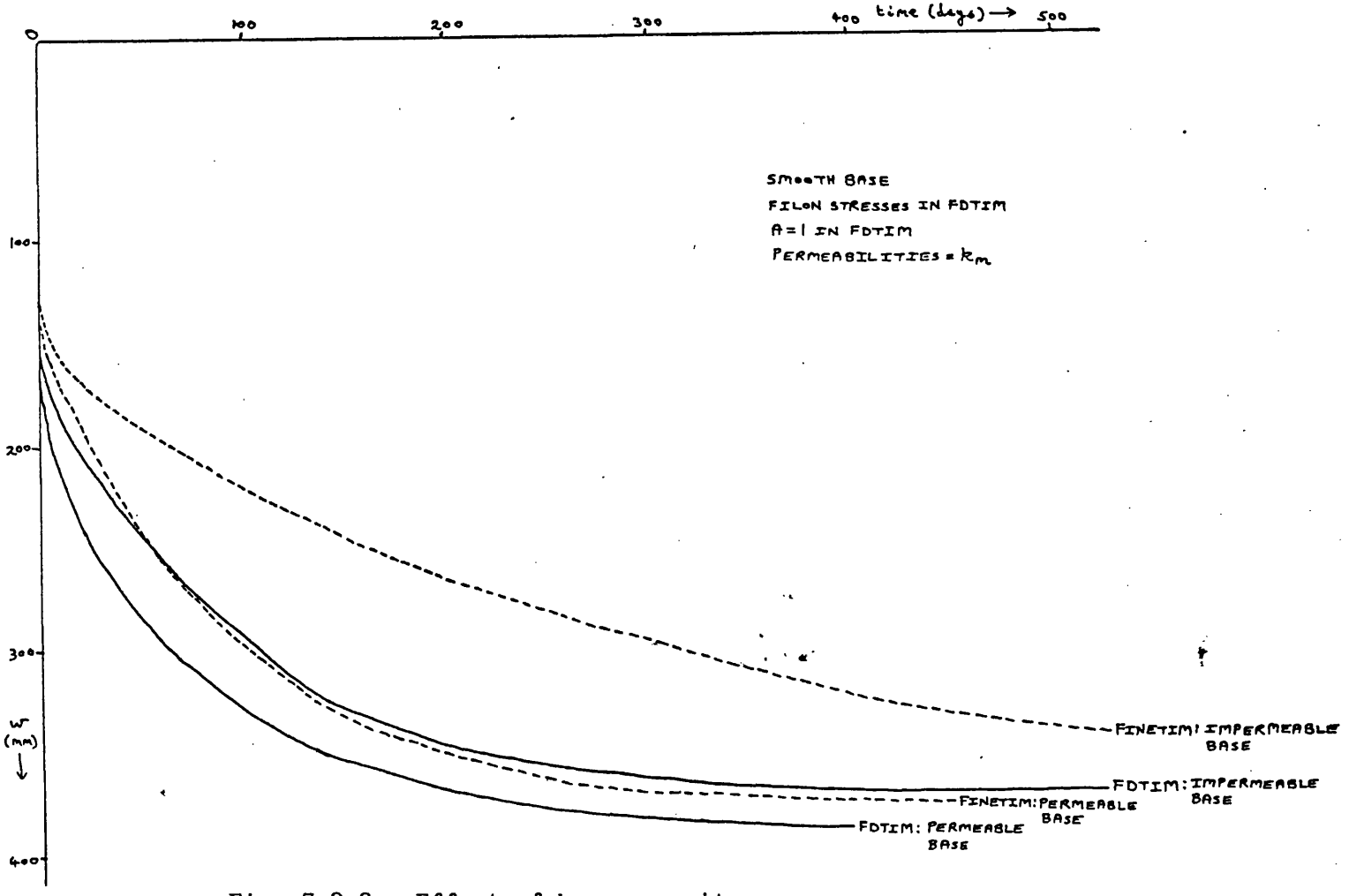


Fig. 7.2.8 Effect of base porosity

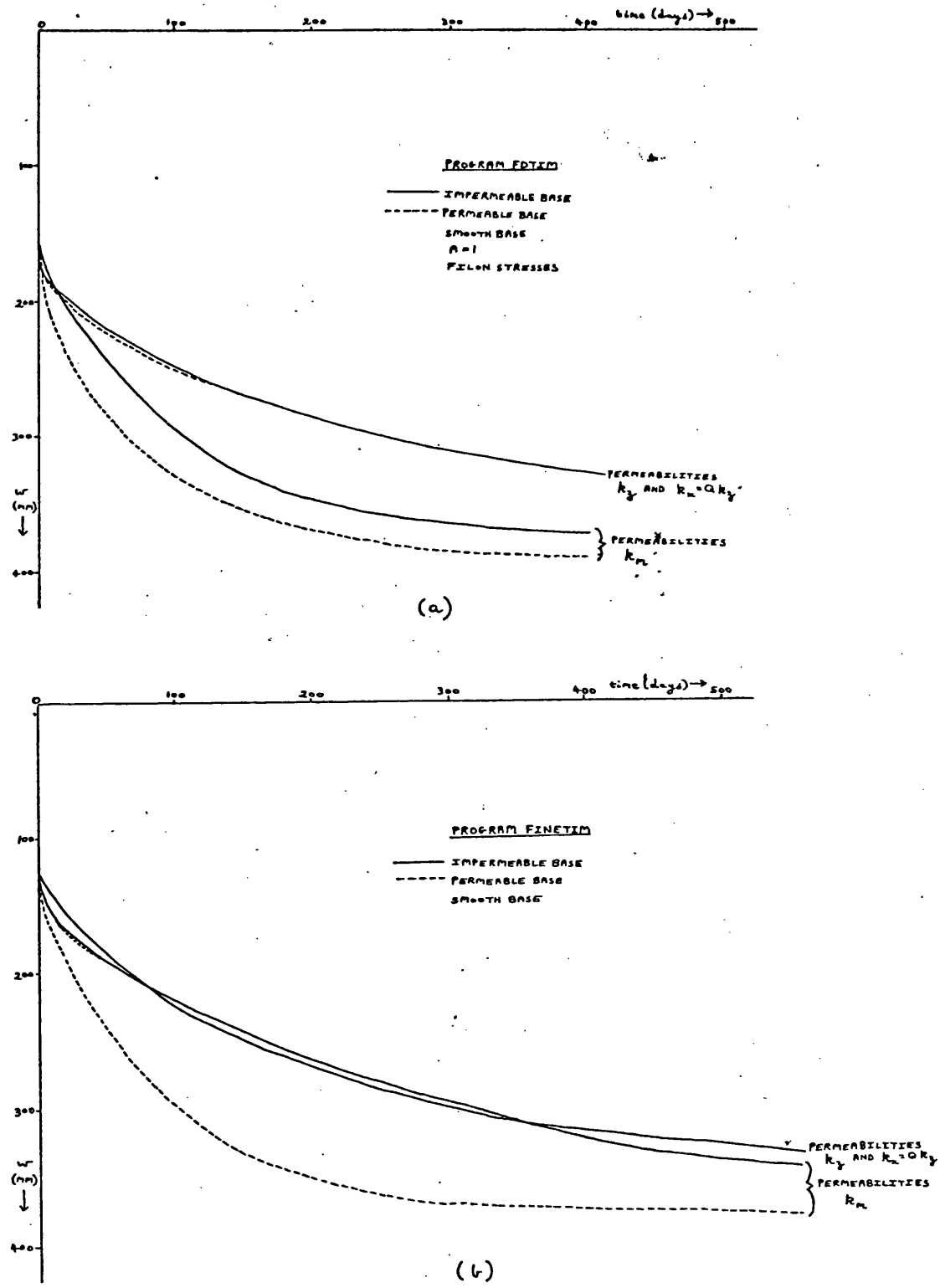


Fig. 7.2.9 Effect of permeability ratios

Comparison with recorded settlements

The predicted settlements quoted above are 50-100% lower than those recorded. Analyses were performed using the hyperbolic stress-strain model for the soil, with the values of cohesion and angle of friction quoted in fig 7.2.2; the loading was performed in five increments. Both programs required extremely small timesteps, and it was discovered that the soil elements at the base of the layer failed before the full load had been reached. The hyperbolic stress-strain law is not a realistic model of soil behaviour at stresses approaching failure. Reasonable results were obtained using FINETIM with arbitrarily increased values of soil cohesion, however.

7.3 Two-bay frame

We stated earlier that a series of rigid footing tests was carried out in the School of Architecture and Building Engineering; this included a soil-structure interaction test using three footings linked by a steel beam (a sketch of the experiment appears in fig 6.3.1). Unfortunately, because of the experimental defects mentioned in §6.3 the results are not considered sufficiently meaningful to warrant comparison with numerical predictions. We will therefore use the experimental layout as a basis on which to study the effect of varying certain parameters, using the programs FDTIM and FINETIM.

The general form of the problem is shown in fig 7.3.1. The soil modulus and all-round permeability are the values determined for the experimental clay (see §6.3), but the Poisson's ratio has been set to zero to maximize the degree of consolidation settlement. We shall use the linear elastic soil model, and take the Skempton pore-pressure coefficient A as $^1/_3$ in FDTIM. The out-of-plane thickness of the experiment is 151mm, and loads of P_1 and P_2 kg are applied to the beam at the points indicated, causing settlements of the footings A, B and C (reading from left to right in fig 7.3.1).

The effects on the footing settlements of

(a) variation of beam stiffness,

(b) variation of boundary conditions,

(c) variation of permeabilities,

as predicted by FDTIM and FINETIM, will be presented below. The finite element mesh used for all analyses is shown in fig 7.3.2. It has 616 nodes and 186 elements; this size approaches the maximum normally possible on the ICL 2980 machine being used. (The limit is imposed by the amount of space required for the three scratch files used in the frontal solution - see appendix B for details.) A mesh of 11 rows and 65 columns was used with FDTIM, the meshpoints largely coinciding with the soil nodes in the finite element mesh. In FDTIM we use the facility for modelling rigid footings (see §2.2), with the parameter m = 2.0. Horizontal dimensions of the model are as in the experiment. The soil depth in the experiment was 75mm; this is used in the analyses below, except where in section (c) the soil depth is trebled.

(a) Variation of beam stiffness

In this section we have taken all soil boundaries to be permeable, the soil depth to be 75mm, and the stress distribution in FDTIM to be given by the finite-layer FILON subroutine. The soil permeabilities k_x and k_z are equal to the all-round permeability k_m. We consider the loads

$$P_1 = 14\text{kg} \qquad P_2 = 2\text{kg}$$

to be instantaneously applied to the beam. Fig 7.3.3 shows the consolidation settlements of the three footings, as predicted by FINETIM, for the experimental beam stiffness ($EI = 0.017\text{KN/m}^2$), and for a beam with one fifth of that stiffness. Also shown are the settlements if no interaction effect is considered; this is done by making the beam stiffness very small and applying loads R_A, R_B, R_C directly to the footings, where

$$\begin{aligned} R_A &= {}^1/_{32}(13P_1 - 3P_2) \\ R_B &= {}^{11}/_{16}(P_1 + P_2) \\ R_C &= {}^1/_{32}(13P_2 - 3P_1). \end{aligned} \qquad (7.3.1)$$

These equations are derived by a structural analysis assuming no differential settlement between the footings.

As would be expected, an increase in the beam stiffness reduces the differential settlements between footings. The same holds when using FDTIM, as shown in fig 7.3.4, but here there are two serious drawbacks, both caused by the iterative procedure used to harmonize footing settlements with footing loads at each timestep (see §3.5). Firstly, it can be seen in fig 7.3.4 that compared to the case where there is no interaction, all the footings settle more rapidly; physically, this should only occur with footings A and C. Secondly, the stiffer the beam the more iterations are required to determine compatible footing settlements and loads. Each iteration requires a re-evaluation of the stress distribution, which takes a large part of the program run-time. Even for the very low stiffnesses used in fig 7.3.4 ($^1/_5$ and $^1/_{25}$ of the experimental stiffness) which required 2-5 iterations at each timestep, the run-times using FDTIM were three times as long (900 secs against 300 secs) as a comparable finite element analysis. As the beam stiffness is further increased, the number of iterations needed rises sharply. For the experimental beam stiffness, initial convergence (to an error of 0.005mm) had not been reached after 29 iterations and 1000 secs of run-time.

(b) Variation of boundary conditions

A study of the effect of base conditions on the footing settlements was made using FINETIM. The model was the same as in the previous section, but with a beam stiffness EI of $0.0033 KN/m^2$, and with the instantaneous loading

$$P_1 = 10kg \qquad P_2 = 5kg.$$

Fig 7.3.5 shows the footing settlement curves for the cases

(i) smooth impermeable base
(ii) rough impermeable base
(iii) smooth permeable base
(iv) no interaction; smooth impermeable base.

Compared to case (i), the rough base in case (ii) reduced the immediate settlements of footings A and B by 20%, although footing C settled slightly more. A permeable base accelerated consolidation.

(c) Variation of permeabilities

The model of section (b) above (with a smooth impermeable base) was also used to observe the effect of a change in the ratio Q of horizontal to vertical permeability (see §7.2). To do this, the vertical permeability was divided by 5 and the horizontal permeability multiplied by 5, thus making $Q = 25$ while keeping constant the all-round permeability. The resulting settlements are compared with those for the isotropic case in fig 7.3.6. All the settlement rates are reduced, indicating that the primary direction of pore-water flow is vertical; however, while footing C (which bears the least load) has its settlement reduced by 30% at time $t = 2100$ secs, the middle footing B has a settlement reduction of only 3%. Settlement of footing A is reduced by 8%. This discrepancy must be due to the pore-water under footing B, which has the largest immediate pressure rise, flowing primarily horizontally toward the lower-pressure area under footing C and retarding the dissipation of pore-pressure there.

The same effect, this time with an actual rise of footing C during consolidation was observed using both FDTIM and FINETIM on a deep soil layer model. For this model, the layer depth was trebled to 225mm (this is easily effected by use of a depth scaling factor ZSCAL written into the programs), with a smooth permeable base but impermeable footings. The infinite-depth Boussinesq stress distribution theory was used in FDTIM. A very flexible beam ($EI = 0.0017 KN/m^2$, one hundredth of the experimental stiffness) was loaded with

$$P_1 = 14kg \qquad P_2 = 2kg$$

and settlement curves were obtained for the permeability ratios $Q = k_x/k_z = 1$, 25 and 0.04 (while keeping the all-round permeability constant). The results from FDTIM and FINETIM are

shown in figs 7.3.7 and 7.3.8 respectively. Both sets of results show similar qualitative behaviour of the footings. From it we conclude that for impermeable footings on deep soil layers, the flow of pore-water under the footings immediately after loading is primarily horizontal, but that after this initial dispersal the consolidation proceeds mainly through vertical dissipation of the excess pore-pressure.

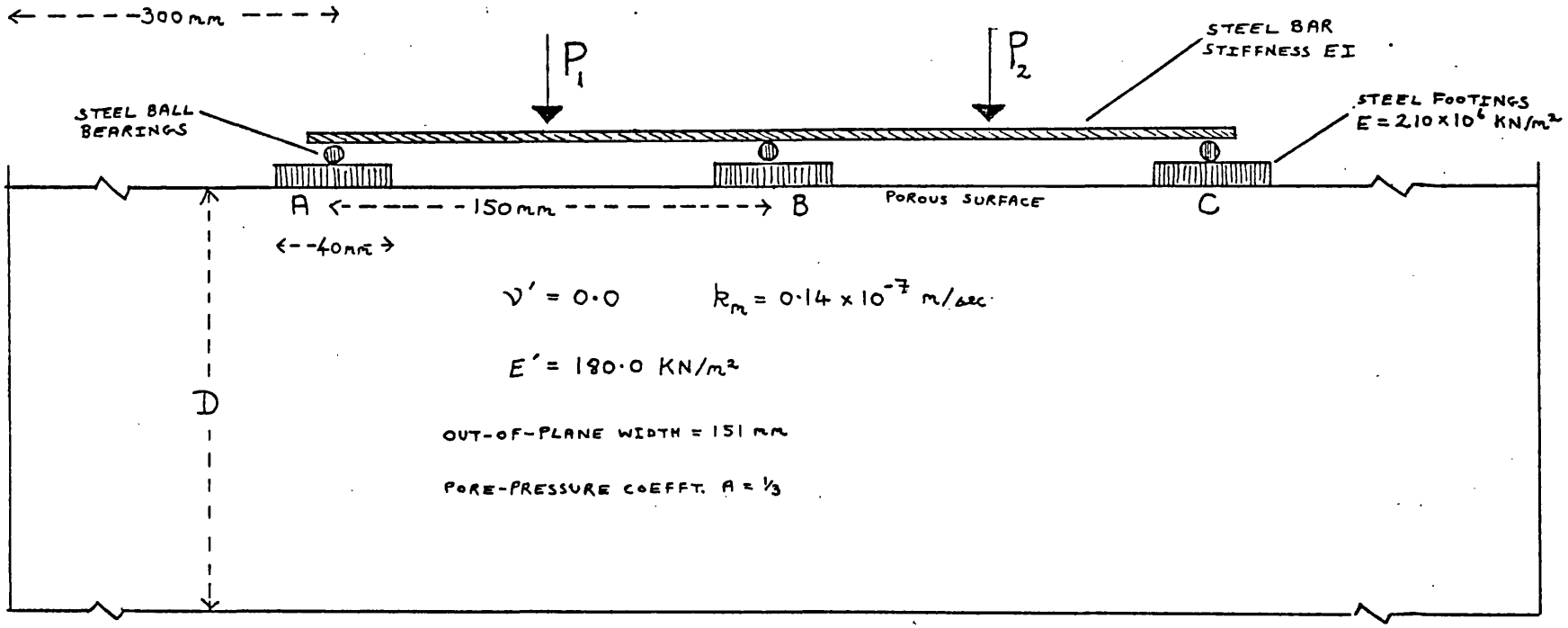


Fig. 7.3.1 Basic two-bay frame model

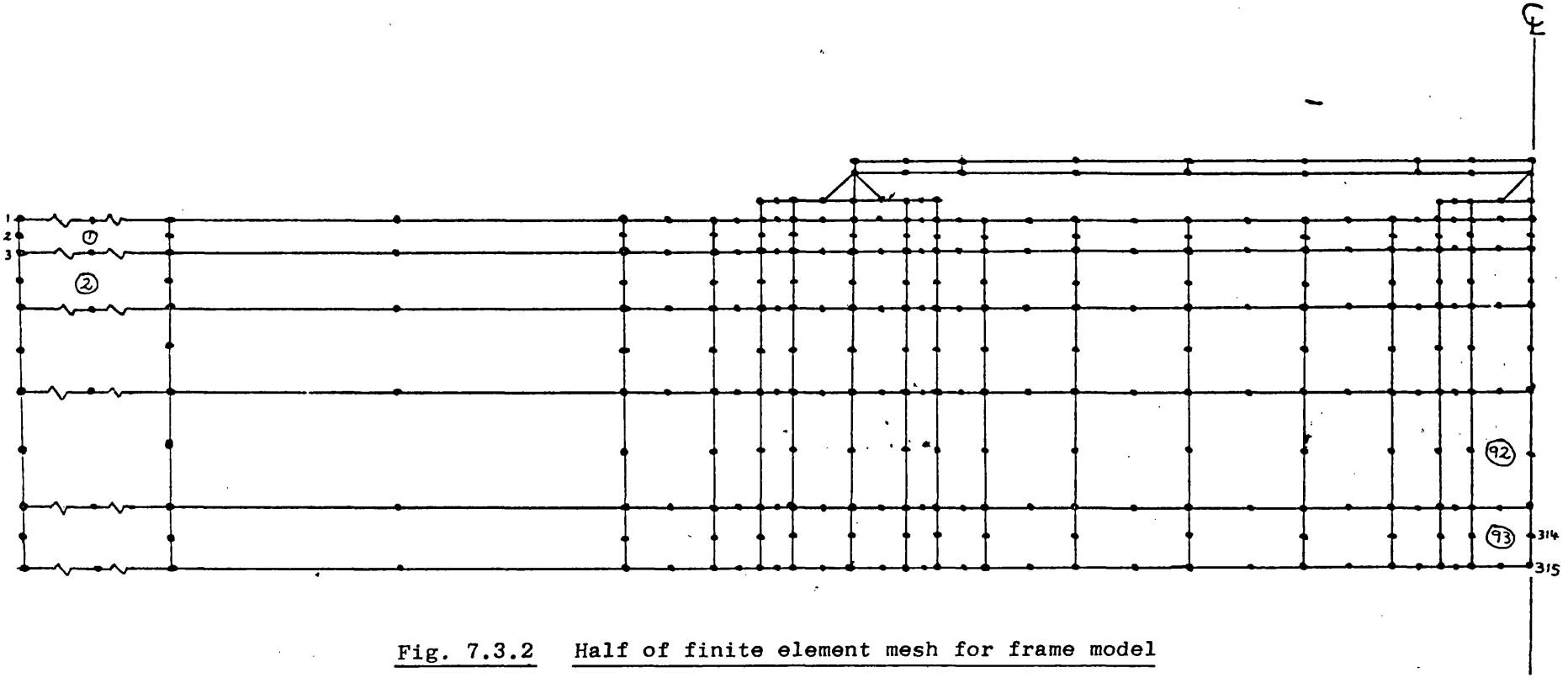


Fig. 7.3.2 Half of finite element mesh for frame model

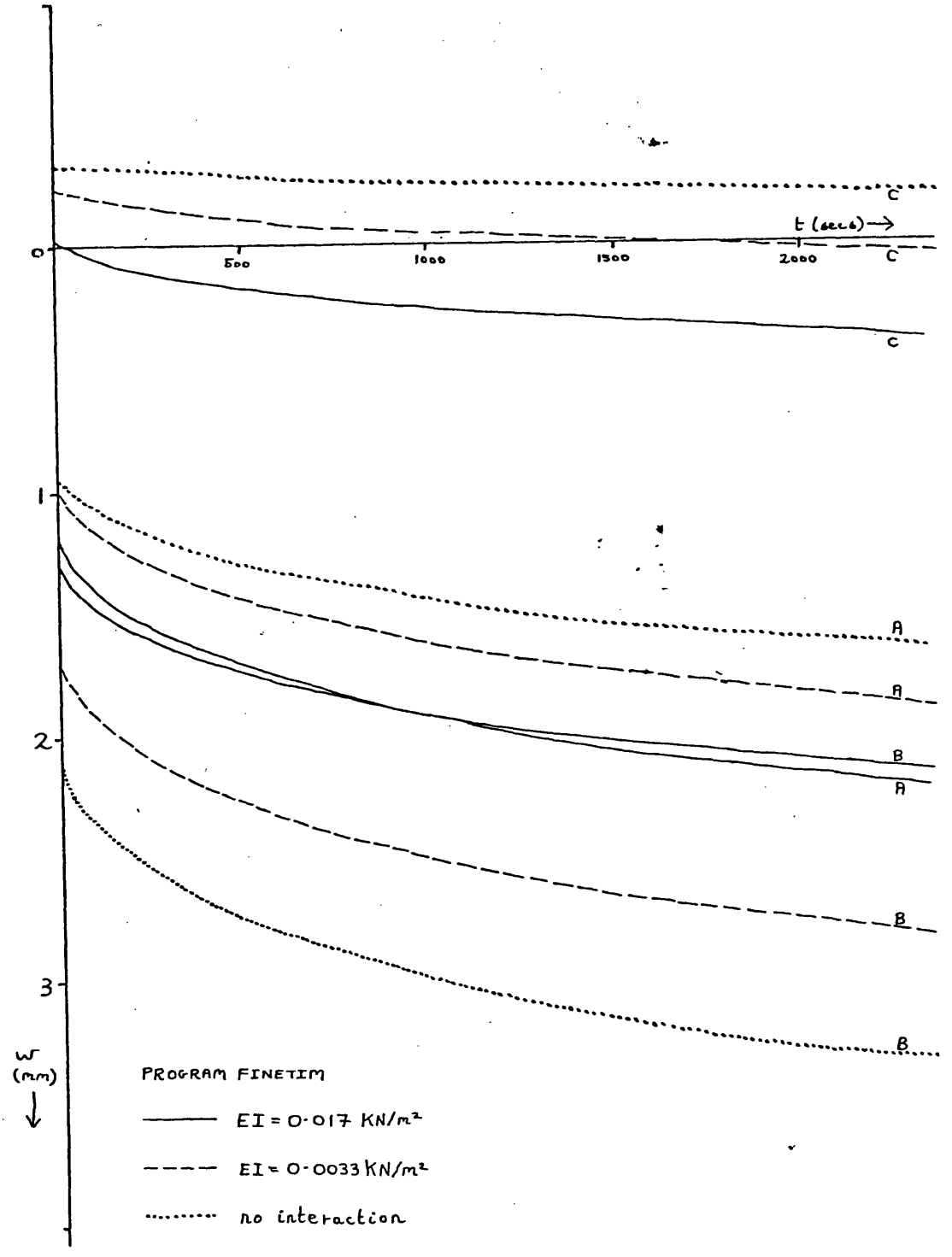


Fig. 7.3.3 Effect of beam stiffness on settlements: FINETIM

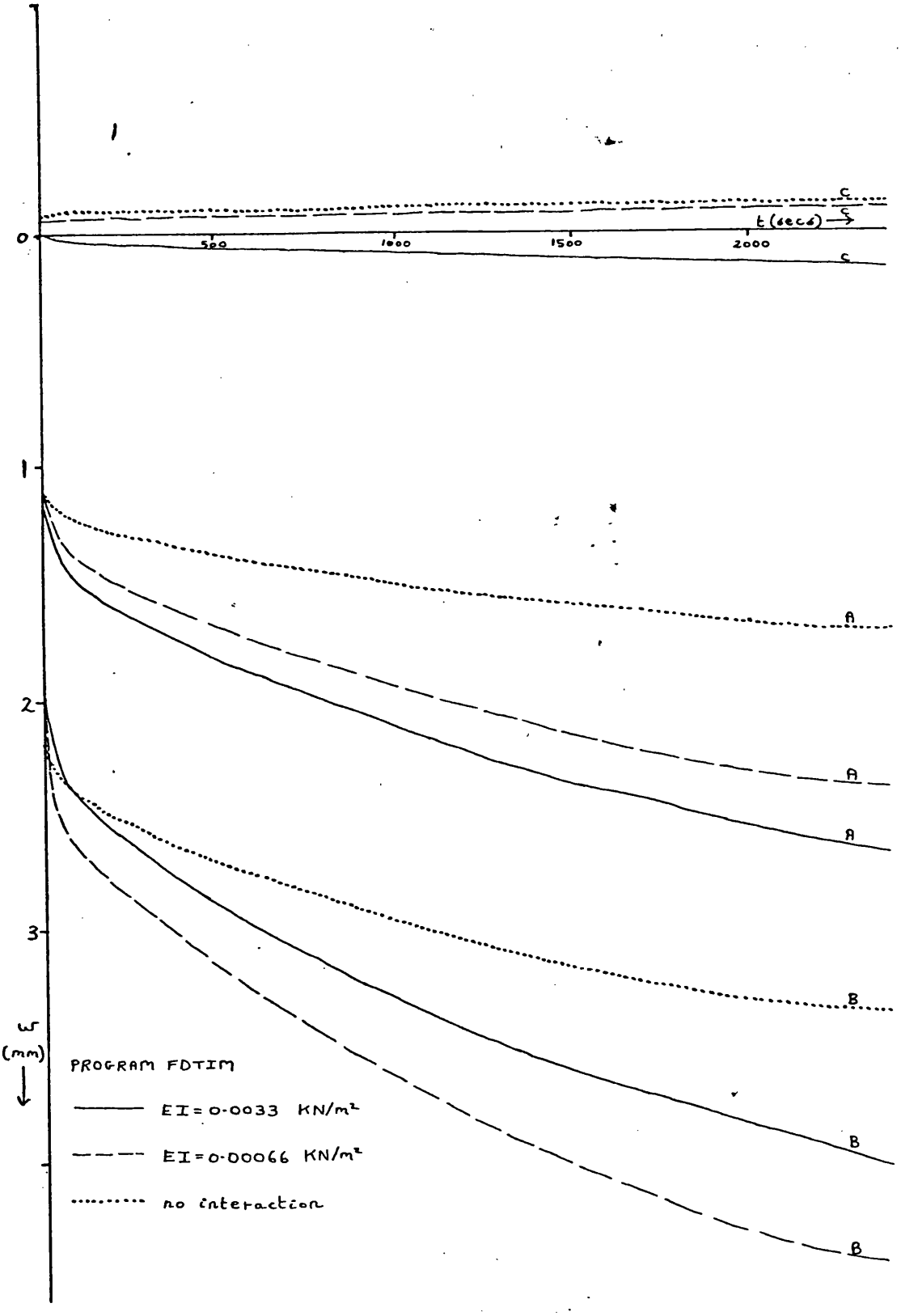


Fig. 7.3.4 Effect of beam stiffness on settlements: FDTIM

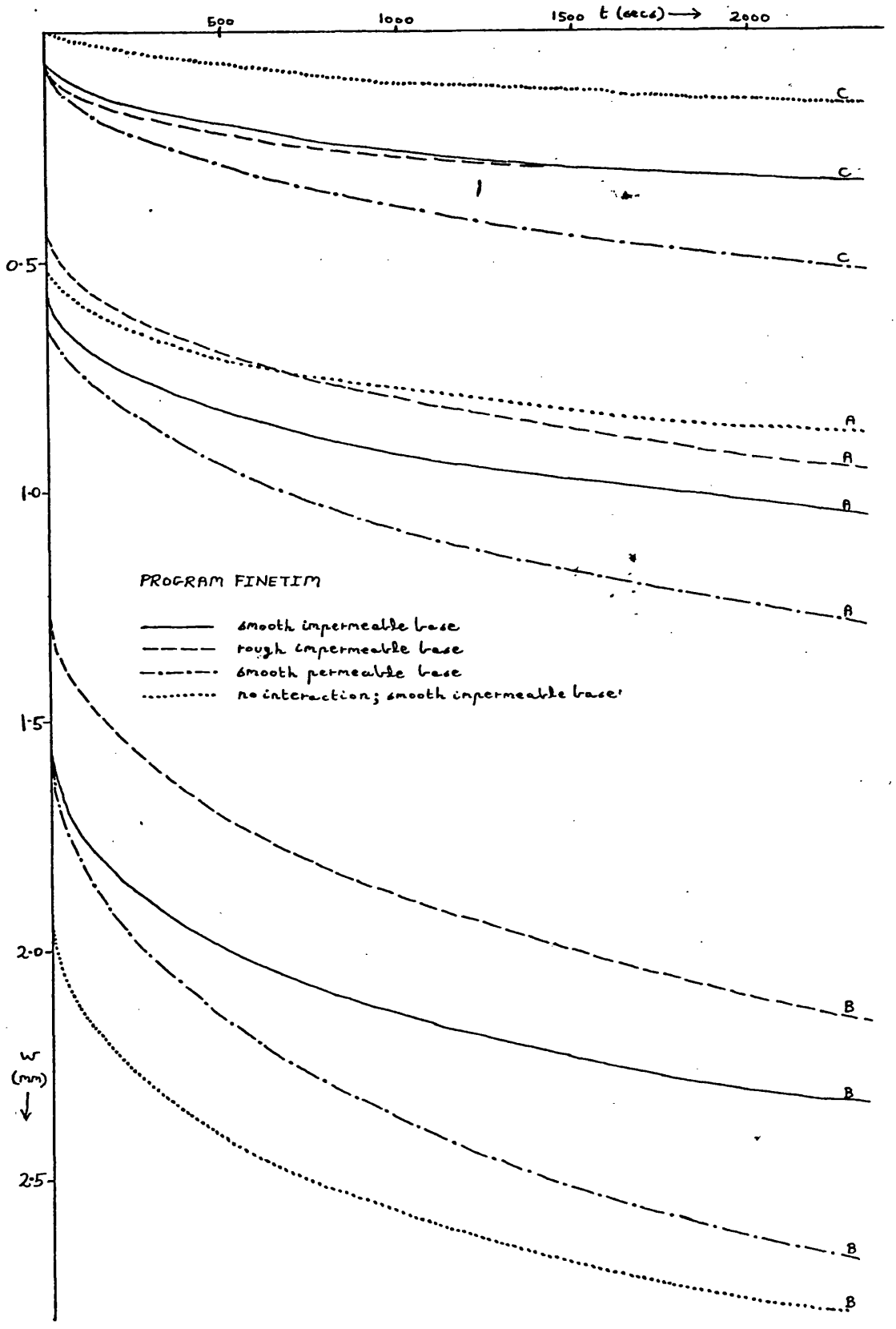


Fig. 7.3.5 Effect of base conditions

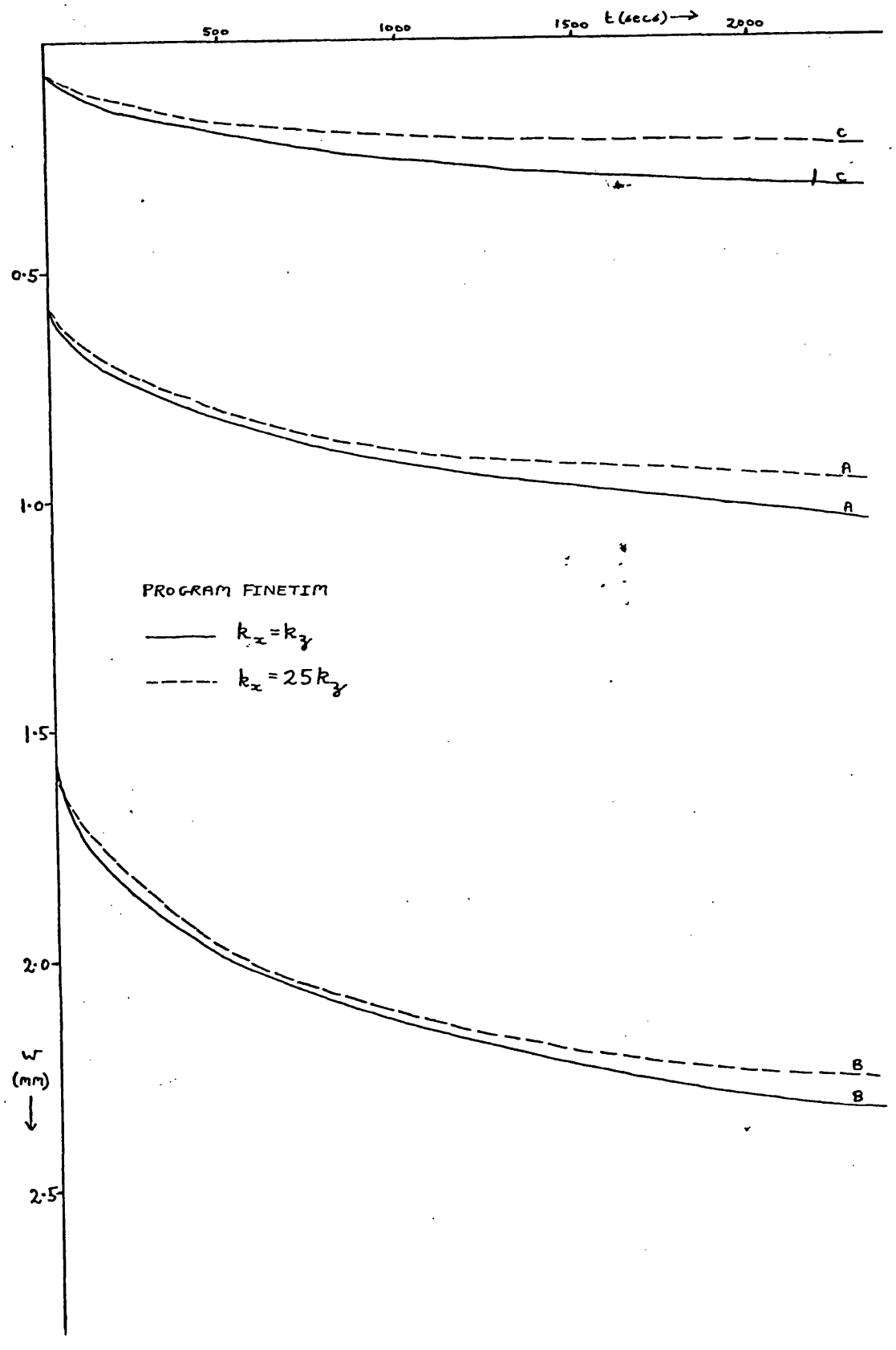


Fig. 7.3.6 Effect of permeability ratio: FINETIM

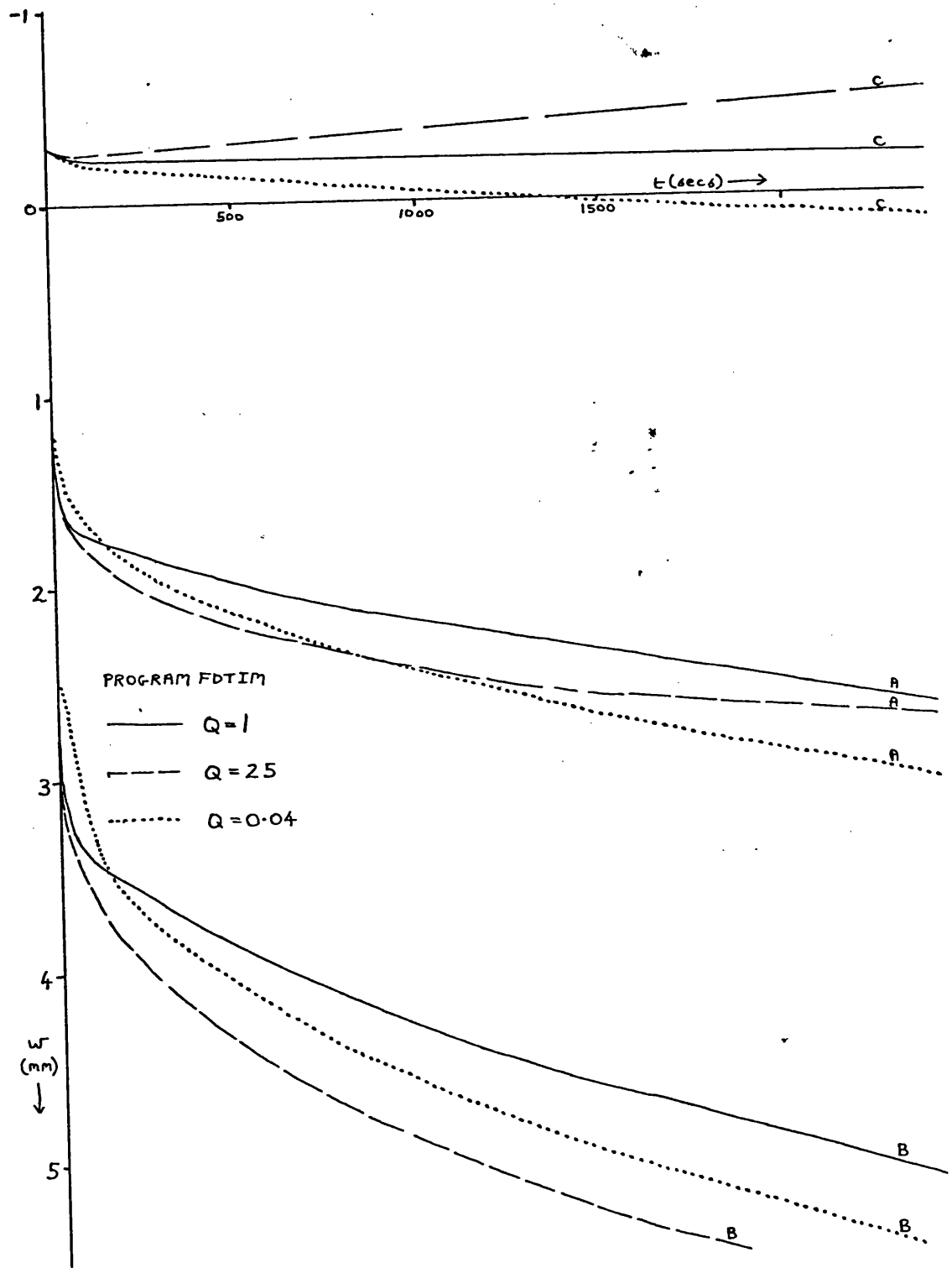


Fig. 7.3.7 Effect of permeability ratio: FDTIM

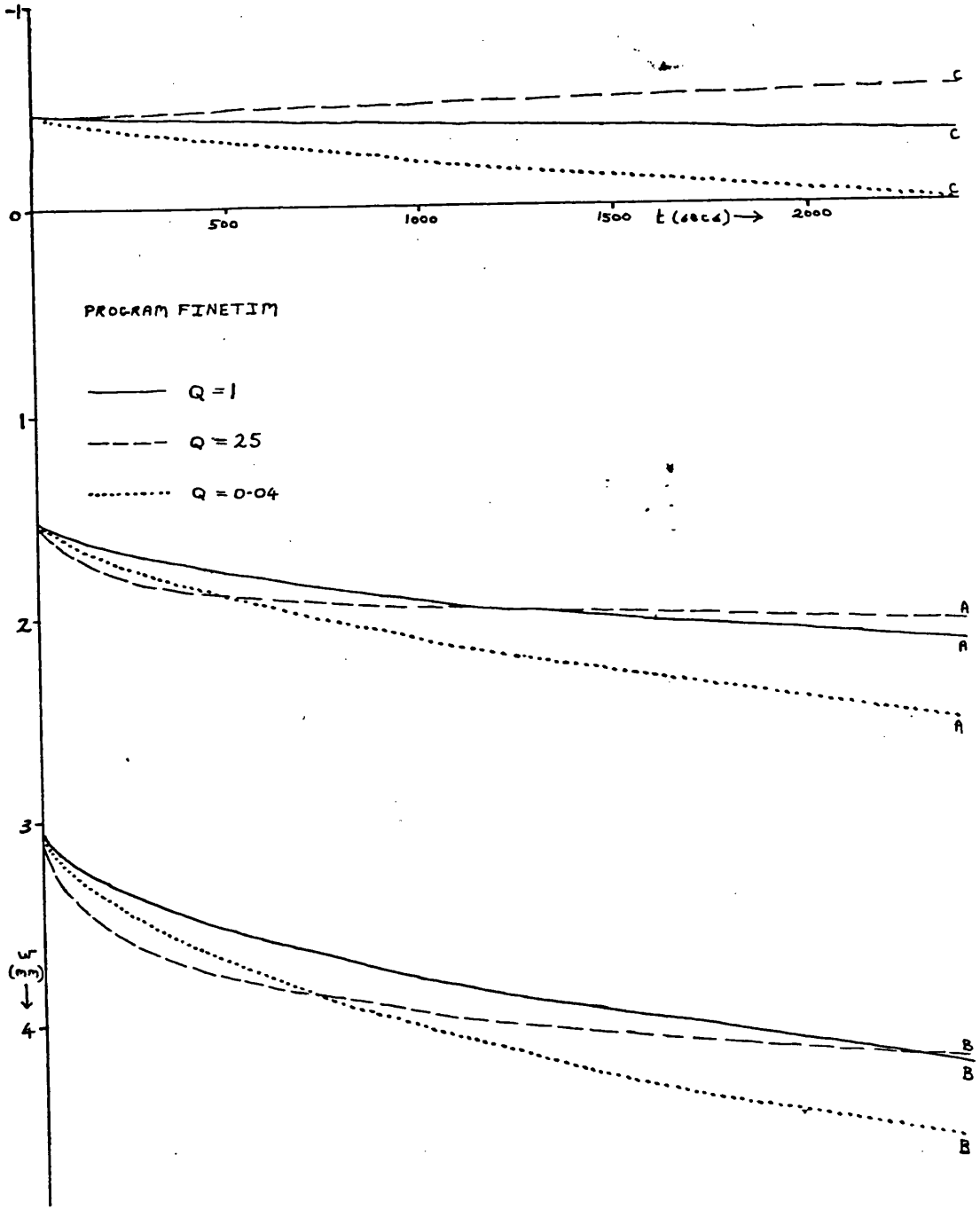


Fig. 7.3.8 Effect of permeability ratio: FINETIM

7.4 Conclusions

It has been seen in §7.2 that the introduction of internal soil layering greatly reduces the efficiency of the finite difference method. This explains the long run-times needed by the two-dimensional consolidation program of Murray (1974) - of the order of 2000 secs. These times are needed even with a coarse mesh, since his program is dimensioned for a maximum of 30 horizontal rows. Further, the stress distribution given by Boussinesq theory has been found to be inaccurate in the embankment problem due to the shallowness of the soil layer and the roughness of the base. The finite element analysis still took considerably longer than that using FDTIM (260 secs against 55 secs), but if a reliable prediction is required for an expensive construction project the extra cost would be worthwhile. If a cheap approximate solution is acceptable, a one-dimensional consolidation solution using finite differences would be considerably simpler than FDTIM and would give virtually identical results. Murray (1972) describes such a program for embankment problems.

We have found in §7.3 that the horizontal flow of pore-water may have a significant effect on the settlement of a framed structure. For such a problem the unified finite element method becomes the most efficient as well as the most realistic model of the problem. It allows great generality in the type of structure modelled, and automatically includes the effect of moments applied to the soil surface by the foundation. The run-time is largely independent of the structural stiffness and the boundary conditions, and it has a clear extension to true three-dimensional consolidation.

CHAPTER 8 - Conclusions and Recommendations for Future Work

In this thesis we have developed numerical techniques for predicting the two-dimensional consolidation settlement of a soil layer under a variety of surface loadings. Two separate methods of solution have been investigated. The first method combines the theory of stress distribution described in Chapter 2 with a numerical solution of the Terzaghi consolidation equation by finite differences, leading in Chapter 3 to the computer program FDTIM. The second approach uses finite elements, and is covered in Chapter 4.

The theoretical developments contained in this thesis may be summarized as follows:

1. The adaptation in § 2.2 of the finite layer stress distribution theory of Filon (1903) into a computer subroutine providing a cheap alternative to the Boussinesq theory commonly used, and giving a considerably more accurate representation of the stresses in a relatively shallow or medium-depth soil layer on a smooth base (see § 6.2).

2. The evaluation of the stress distribution from a loaded area by a method of discretization into line-loads. This method allows any sort of contact stress profile across the loaded area to be used, and this versatility has been taken advantage of in modelling rigid loaded footings. The original contact stress profile for a rigid footing on a half-space by Borowicka (1939) has been adapted to model the case of a finite soil layer. This is expressed in equation 2.2.19, which incorporates a parameter m dependent on relative layer depth.

   A reduction in layer depth will reduce the differential settlement occurring between the centre and edge of a flexible loaded footing, as will the introduction of a degree of rigidity in that footing. This suggests that the above adaptation might also be used to model a footing of finite stiffness, ie. a raft foundation. In this case, the parameter m would be chosen according to the raft stiffness as well as the relative layer depth; it could be related to these factors by comparison with the theoretical contact stress profiles of Brown (1969 a and b).

3. A simple form of finite element analysis of a beam on individual footings (see §3.4), which derives a matrix relating applied loads and footing displacements to footing reactions and beam deformations. Combined with the equivalent stiffness theory of Meyerhof (1953) it can provide a fast simplified analysis of a framed structure.

4. The incorporation (in § 4.3 - 4) of the theory of consolidation by finite elements (Sandhu and Wilson (1969)) into a general finite element program, in such a way that structural as well as soil elements may be used in the mesh.

5. The further generalization (in § 4.3) of this theory by the introduction of a smoothing process which allows normal types of element to be used in a mesh, instead of only the special 'composite' soil elements so far employed for consolidation analyses.

6. The application (in § 5.2) of the mathematical theory of smoothing splines to the analysis of compression test data, to estimate preconsolidation pressure.

Results from the use of the two computer programs in a number of loading problems are given and compared in Chapters 6 and 7. In Chapter 6 we see that the finite difference program FDTIM can accurately predict the consolidation settlements of flexible and rigid footings on homogeneous linear elastic soil layers, requiring only about a quarter of the run-time for a comparable finite element analysis. On the other hand, we find in § 7.2 that the introduction of complexities such as soil layering and a non-linear soil model greatly increases the cost of the finite difference method of solution. An even greater cost increase in FDTIM is caused by the introduction of a frame structure linking the footings, as in § 7.3, and the run-time becomes more than three times that for a finite element analysis using FINETIM. Because of this and other limitations of FDTIM in the frame problem (discussed in § 7.4), the writer concludes that the most efficient and satisfactory approach to two-dimensional consolidation problems

involving layered soils, non-linear elastic soils or soil-structure interaction effects is a unified finite element analysis; the program FINETIM is a very general program of this type. Its disadvantages are in the amounts of input data and computer storage required. By far the largest part of the input data describes the mesh geometry - that is, the node numbers around each element and the coordinates of each node. If a rectangular mesh of eight noded quadrilateral elements is to be used for the soil, such data could be generated by a short separate computer program taking as input the x - and z - coordinates of the mesh columns and rows respectively. Numbering of the nodes and elements could start from the bottom left-hand side of the mesh and proceed horizontally, so that any structural elements and nodes above the soil surface could be added by the user at the end of the computer-generated data, followed by the remainder of the input data. Storage space may be reduced by using a coarse mesh having fewer elements, but using the quadratic-elements-with-smoothing process developed in § 4.3; this gives the same detail regarding nodal pore-pressures as a composite element mesh with four times the number of elements.

The extension of the anlysis to three-dimensional problems is of great importance in practical applications. In this thesis we have considered plane strain problems only. As was pointed out in §4.5, however, the finite element program FINEPAK on which FINETIM is based is designed to cope with axisymmetric and three-dimensional problems also, and in the adaptations made by the writer this generality has been maintained (the only exception being the two-dimensional pore-pressure smoothing subroutine). There is now a need for a systematic examination of the performance of FINETIM on three-dimensional frame and raft problems to be made, comparing it with Wood's (1972) program (which uses Boussinesq stress distribution theory and a finite difference solution of the one-dimensional consolidation equation) and if possible with results from model tests and site records. It would be very expensive in terms of computer time to extend a finite difference program to handle three-dimensional flow, and it is therefore important to establish under what circumstances the horizontal flow of pore-water can significantly affect settlements.

There are a number of refinements which could be incorporated in the

programs. The stiffness of the structural elements could be increased with time to represent the construction process, addition of cladding etc. This could be achieved in FINETIM without reompiling the element stiffness matrices, in the same way as the hyperbolic stress-strain model has been implemented (see § 5.3). Wood (1972) has concluded that this factor has only a small effect on the deformations. We have described the pore-pressure boundary conditions for a perfectly porous and a completely impermeable boundary; in reality the porosity condition would lie somewhere between these extremes. Such a partially porous condition is given by:

$$\frac{\partial u}{\partial n} = \theta u \qquad 0 < \theta < \infty \qquad (8.1)$$

where $\frac{\partial u}{\partial n}$ is the derivative of pore-pressure normal to the boundary. The condition has recently been used in an analytic study by Deresiewicz (1979); it could be included routinely in a finite difference scheme, but would need special provisions in a finite element program.

A more important theoretical refinement of FINETIM would be the provision for loss of contact and relative horizontal slipping between structure and soil. This would require the use of special one-dimensional interface elements (see, for example Desai and Christian (1977), Chapter 4). A means of taking advantage of the symmetry of a problem by modelling only half of it in FINETIM would improve efficiency (simple use of the natural boundary conditions for pore-pressures down the centreline has been found to give incorrect results there).

Finally, the writer would recommend the incorporation into FINETIM of a simple form of critical state soil model, such as that described by Burland (1972)

BIBLIOGRAPHY

Note: References marked with an asterisk, eg (1967)*, are secondary references which have not been consulted directly.

ABBOTT, M B (1960)* "One-Dimensional Consolidation of Multi-Layered Soils", Geotechnique 10, pp 151-165

BIOT, M A (1935)* "Le Problème de la Consolidation des Matières Argileuses sons une Charge", Annaies de la Societé Scientifique de Bruscelles, Series B.55

BLOSS, J W (1976) "An investigation into applications of alternating direction implicit methods to partial differential equations arising from a heat diffusion problem in the study of breast cancers", MSc dissertation, Univ of Bath

BOOKER, J R (1973) "Consolidation of a Layer under a Strip Load", J Soil Mech Found Div ASCE, SM4 (1973) pp 358-360

BOOKER, J R and SMALL, J C (1975) "An investigation of the stability of numerical solutions of Biot's equations of consolidation", Int J Solids Struct, vol 11, pp 907-917

BOOKER, J R and SMALL, J C (1977) "Finite Element Analysis of Primary and Secondary Consolidation", Int J Solids Structures, vol 13, pp 137-149

BOROWICKA, H (1938) "The distribution of pressure under a uniformly loaded elastic strip resting on elastic-isotropic ground", Second Cong Intem Assoc Bridge and Structural Eng, Final Report, VIII 3, Berlin

BOROWICKA, H (1939)* "Druckverteilung unter Elastischen Platten", Ingenieur Archiv, Vol X, No 2, pp 113-125

BOUSSINESQ, J (1885)* "Application des Potentiels à l'Etude de l'Equilibre et du Mouvement des Solides Elastiques", Paris, Gauthier-Villard

BROWN, C B and KING, I P (1966)* "Automatic embankment analysis - Equilibrium and instability conditions", Geotechnique, vol 16, pp 209-219

BROWN, P T (1969a) "Numerical Analysis of Uniformly-Loaded Circular Rafts on Elastic Layers of Finite Depth", Geotechnique 19, No 2, pp 301-306

BROWN, P T (1969b) "Numerical Analysis of Uniformly-Loaded Circular Rafts on Deep Elastic Foundations", Geotechnique 19, No 3, pp 399-404

BURLAND, J B (1967)* "Deformation of Soft Clay", PhD thesis, Cambridge University

BURLAND, J B (1972) "A method of estimating the pore-pressures and displacements beneath embankments on soft, natural clay deposits", in "Stress-strain behaviour of soils", pp 505-536, Foulis

BURLAND, J B and WROTH, C P (1975) "Settlement of Buildings and Associated Damage" in "Settlement of Structures", pp 611-654, Pentech Press, London

BURMISTER, D M (1956)* "Stress and Displacement characteristics of a two-layer rigid base soil system: influence tables and practical applications", Proc Highw Res Bd, vol 35, p 773

CASAGRANDE, A (1936)* "The determination of the preconsolidation load and its practical significance", Proc 1st Int Conf Soil Mech Found Eng, p 60

CHAMECKI, S (1956)* "Structural Rigidity in Calculating Settlements", Jnl Soil Mech Fdns Div ASCE, vol 82, SMI, pp 1-19

CHRISTIAN, J T and BOEHMER, J W (1970) "Plane Strain Consolidation by Finite Elements", J Soil Mech Found Div ASCE, vol 96, pp 1435-1457

CHRISTIAN, J T; BOEHMER, J W and MARTIN, P P (1972)* "Consolidation of a Layer under a Strip Load", J Soil Mech Found Div ASCE, vol 98, SM7, pp 693-707

CROWSER, J C; SCHUSTER, R L and SACK, R L (1975) "Settlement and contact pressure distribution of a mat-supported silo group on an elastic subgrade" in "Settlement of Structures", pp 344-352, Pentech Press, London

CRYER, C W (1963) "A Comparison of the Three-Dimensional Theories of Biot and Terzaghi", Jnl Mech Appl Math, 16, pp 401-412

DAVIS, E H and POULOS, H G (1968) "The Use of Elastic Theory for settlement prediction under three-dimensional conditions"; Geotechnique vol 18, pp 67-91

DAVIS, E H and POULOS, H G (1972) "Rate of Settlement under Two- and Three-Dimensional Conditions", Geotechnique 22, No 1, pp 95-114

De JONG, G J and VERRUIJT, A (1965) "Primary and Secondary Consolidation of a Spherical Clay Sample", Proc 6th Int Conf Soil Mech, Montreal, 1, p 254

DERESIEWICZ, H (1979) "Effects of Restricted Flow at the Surface of Saturated Clay", Int J Numerical and Analytical Methods in Geomechanics, vol 3, pp 1-11

DESAI, C S (1971)* "Nonlinear analysis using spline functions", J Soil Mech Found Div ASCE, vol 97, pp 1461-1480

DESAI, C S and ABEL, J F (1972) "Introduction to the Finite Element Method", Van Nostrand Reinhold (New York)

DESAI, C S (1975) "Analysis of Consolidation by Numerical Methods", Proc Gen Sess Symp Recent Dev Soil Mech Univ NSW, pp 143-179

DESAI, C S and CHRISTIAN, J T (1977) ed "Numerical Methods in Geotechnical Engineering", McGraw-Hill (New York)

DUNCAN, J M and CHANG, C Y (1970)* "Non-linear analysis of stress and strain in soils", J Soil Mech Found Div ASCE, vol 96, pp 1629-1653

FENNER, R T (1975) "Finite Element Methods for Engineers", Macmillan, London

FILON, L N G (1903) "On an Approximate Solution for the Bending of a Beam of Rectangular Cross-Section under any System of Load, with Special Reference to Points of Concentrated or Discontinuous Loading", Phil Trans Roy Soc London, vol 201, pp 65-155

FORSYTHE, G E and WASOW, W R (1959) "Finite Difference Methods for Partial Differential Equations", J Wiley and Sons, Inc

GHABOUSSI, J and WILSON, E L (1973) "Flow of compressible fluid in porous elastic media", Int J Num Methods Eng, vol 5, No 3, pp 419-442

GIBSON, R E (1963)* "An analysis of system flexibility and its effect on time-lag in porewater pressure measurements", Geotechnique, vol 13, pp 1-11

GIBSON, R E; KNIGHT, K and TAYLOR, P W (1963)* "A Critical Experiment to examine Theories of Three-Dimensional Consolidation", Proc Europ Conf Soil Mech, Wiesbaden, 1, pp 69-76

GIBSON, R E; SCHIFFMAN, R L and PU, S L (1970) "Plane strain and axially symmetric consolidation of a clay layer on a smooth impervious base", Quart Journ Mech and Applied Math, Vol XXIII, Pt 4, pp 505-520

GIBSON, R E and SILLS, G C (1972) "Some results concerning the plane deformation of a non-homogeneous elastic half-space" in "Stress-strain behaviour of soils", pp 564-572, Foulis

GIBSON, R E (1974) "The analytical method in soil mechanics", Geotechnique, vol 24, pp 115-140

GREEN, P A and HIGHT, D W (1975) "The failure of two oil-storage tanks caused by differential settlement", in "Settlement of Structures", Pentech Press, London, pp 353-360

GREVILLE, T N E ed (1969) "Theory and Applications of Spline Functions", Academic Press (New York)

HADDADIN, M J (1971)* "Mats and combined footings -- analysis by the finite element method", Proc Am Concrete Inst, vol 68, pp 945-949

HAGMANN, A J (1971)* "Prediction of stress and strain under drained loading conditions", Tech Dept Civ Eng Rep R71-3

HAIN, S J and LEE, I K (1974) "Rational analysis of raft foundation", J Geotech Div ASCE, vol 100, pp 843-860

HARR, M E (1977) "Mechanics of Particulate Media", McGraw-Hill, New York

HENKEL, D J (1958)* "Correlation between deformation, pore water pressure, and strength characteristics of saturated clays", thesis, PhD in Engineering, Imperial College of Science and Technology, London

HENKEL, D J (1960)* "The Shear Strength of Saturated Remoulded Clays", Proc Am Soc Civil Engnrs, Res Conf Shear Strength Cohesive Soils, p 533

HINTON, E and CAMPBELL, J S (1974) "Local and Global Smoothing of Discontinuous Finite Element Functions using a Least Squares Method", Int J Num Meth Eng, vol 8, pp 461-480

HINTON, E and OWEN, D R J (1977) "Finite Element Programming", Academic Press

HOOPER, J A (1973) "Observations on the behaviour of a piled-raft foundation on London Clay", Proc ICE, vol 55, pp 855-877

HORNE, M R and MERCHANT, W (1965) "The Stability of Frames", Pergamon Press

HWANG, C T; MORGENSTERN, N R and MURRAY, D W (1971) "On Solutions of Plane Strain Consolidation Problems by Finite Element Methods", Can Geotech J, 8, pp 109-118

INSTITUTION OF STRUCTURAL ENGINEERS (1978) "Structure-soil interaction: a state of the art report", London

IRONS, B M (1970)* "A Frontal Solution Program", Int J Num Meth Eng, vol 2, pp 5-32

JAKY, J (1944)* "The Coefficient of Earth Pressure at Rest", J Soc Hungarian Architects and Engineers, pp 355-358

JUMIKIS, A R (1969) "Theoretical Soil Mechanics", Van Nostrand Reinhold, New York

KING, G J W and CHANDRASEKARAN, V S (1975) "An assessment of the effects of inter-action between a structure and its foundation", in "Settlement of Structures", pp 368-383, Pentech Press, London

KING, G J W and CHANDRASEKARAN, V S (1977) "Interactive analysis using a simplified soil model", Proc Int Symp on Soil-Structure Interaction, Roorkee, India

KONDNER, R L (1963)* "Hyperbolic stress-strain response: cohesive soils", J Soil Mech Found Div ASCE, vol 89, pp 115-143

LAMBE, W T and WHITMAN, R V (1969) "Soil Mechanics", J Wiley and Sons, Inc, New York

LARNACH, W J (1970) "Computation of settlements on building frames", Civ Eng, vol 65, pp 1040-1044

LARNACH, W J and WOOD, L A (1972) "The effect of soil-structure interaction on settlements", Int Symp Computer-aided Structural Design, Dept Eng, Univ Warwick

LEE, I K (1975) "Structure-Foundation-Supporting Soil Interaction Analysis", Proc Gen Sess Symp Rec Dev Soil Mech, Univ NSW

LEE, I K and HARRISON, H B (1970) "Structure and Foundation Interaction Theory", J Struct Div ASCE, 96, pp 177-197

LEONARDS, G A ed (1962) "Foundation Engineering", McGraw-Hill, New York

LITTON, E and BUSTON, J M (1968)* "The Effect of Differential Settlement on a large, rigid, steel-framed, multi-storey Building", The Structural Engineer, vol 46, pp 353-356

LO, K Y (1969)* "Pore pressure-strain Relationships of Normally Consolidated Undisturbed Clays. Parts 1 and 2." Canad Geotechnical Journal, 6, pp 383-412

MAJID, K I and CRAIG, J S (1971) "An Incremental Finite Element Analysis of Structural Interaction with Soil of non-linear properties", Proc Midland Soil Mech Fed Eng Soc Conf, Birmingham

MANDEL, J (1953) "Consolidation des Sols (Etude Mathematique)", Géotechnique 3, pp 287-299

MEYERHOF, G G (1947)* "The Settlement Analysis of Building Frames", The Structural Engineer, vol 25, pp 369-409

MEYERHOF, G (1953) "Some Recent Foundation Research and its Application to Design", Structural Engineer, vol 31, pp 151-167

MILOVIC, D M; TOUZOT, G and TOURNIER, J P (1970) "Stresses and displacements in an elastic layer due to inclined and eccentric load over a rigid strip", Geotechnique, vol 20, pp 231-252

MITCHELL, A R and WAIT, R (1977) "The Finite Element Method in Partial Differential Equations", J Wiley and Sons (London)

MORRIS, D (1966)* "Interaction of continuous frames and soil media", Jnl Struct Div Am Soc Civ Engrs, vol 92, pp 13-43

MURRAY, R T (1971) "Embankments constructed on soft foundations: settlement study at Avonmouth", Road Research Laboratory report LR419

NAYLOR, D J (1974) "Stresses in Nearly Incompressible Materials by Finite Elements with Application to the Calculation of Excess Pore Pressures", Int J Num Meth Engng, 8, pp 443-460

NAYLOR, D J and HOOPER, J A (1975) "An effective stress finite element analysis to predict the short- and long-term behaviour of a piled-raft foundation on London Clay", in "Settlement of Structures", pp 394-402, Pentech Press, London

NAYLOR, D J (1977) "FINEPAK (Mark 3): User Instructions and Explanatory Notes", Centre for Numerical Methods in Engineering, Dept of Civil Engineering, University College of Swansea

NAYLOR, D J ed (1978) "Finite Elements in Geotechnical Engineering", course lecture notes, Institute for Numerical Methods in Engineering, University College of Swansea

PEACEMAN, D W and RACHFORD, H H (1955) "The Numerical Solution of Parabolic and Elliptic Differential Equations", J Soc Indust Appl Math, vol 3, pp 28-41

PHILLIPS, G M and TAYLOR, P J (1973) "Theory and Applications of Numerical Analysis", Academic Press (London)

PICKETT, G (1938) "Stress distribution in a loaded soil with some rigid boundaries", Proc Highway Research Board, vol 18(2), pp 35-48

POULOS, H G (1967) "Stresses and Displacements in an elastic layer underlain by a rough rigid base", Geotechnique, vol 17, pp 378-410

POULOS, H G and DAVIS, E H (1974) "Elastic Solutions for Soil and Rock Mechanics", J Wiley

POULOS, H G (1975) "Settlement of Isolated Foundations", Proc Gen Sess Rec Dev Soil Mech, Univ NSW

PRENTER, P M (1975) "Splines and Variational Methods", John Wiley and Sons, New York

PRENTER, P M and RUSSELL, R D (1976) "Orthogonal Collocation for Elliptic Partial Differential Equations", SIAM J Numer Anal, vol 13, pp 923-939

RAYMOND, G T and CHAN, H T (1966)* "The Consolidation of Multi-layered Soils subjected to large load ratios and one-dimensional drainage", Dept of Highways, Ontario, Report RB104

REED, M B (1980) "An HPL program to analyse e-log p curves", Research report, School of Architecture and Building Eng, University of Bath

ROSCOE, K H and BURLAND, J B (1968)* "On the generalized stress-strain behaviour of "wet" clay", in "Engineering Plasticity", pp 535-609, CUP

ROSCOE, K H (1970) "10th Tankine lecture: the influence of strains in soil mechanics", Geotechnique, vol 20, pp 129-170

SANDHU, R S and WILSON, E L (1969) "Finite Element Analysis of Seepage in Elastic Media", J Eng Math Div ASCE, EM3 (1969), pp 641-652

SANDHU, R S; LIU, H and SINGH, K J (1977)* "Numerical Performance of some Finite Element Schemes for analysis of seepage in porous elastic media", Int J Num and Anal Meth in Geomechanics, vol 1, pp 177-194

SCHIFFMAN, R L; CHEN, A T and JORDAN, J C (1969) "An Analysis of Consolidation Theories", Soil Mech Found Div ASCE, 95, pp 285-312

SCHMERTMANN, J M (1955)* "The undisturbed consolidation of clays", Trans Am Soc Civil Engrs, vol 120, p 1201

SCHOENBERG, I J (1964)* "Spline functions and the problem of graduation", Proc Nat Acad Sci USA, vol 52, pp 947-950

SCHOFIELD, A N and WROTH, C P (1968) "Critical State Soil Mechanics", McGraw Hill (London)

SKEMPTON, A W (1954)* "The Pore-Pressure Coefficients A and B", Geotechnique 4, No 4, pp 143-147

SKEMPTON, A W (1960)* Correspondence, Geotechnique 10, No 4, pp 186-187

SKEMPTON, A W and McDONALD, D H (1956)* "The Allowable Settlement of Buildings", Proc Inst Civ Engrs, pp 727-768

SMITH, G D (1965) "Numerical Solution of Partial Differential Equations", OUP (reprinted 1975)

SMITH, I M (1970)* "A finite element approach to elastic soil-structure interaction", Canad Geotech J, vol 7, pp 95-105

SOMMER, H (1965) "A Method for the Calculation of Settlements, Contact Pressures and Bending Moments in a Foundation including the Influence of the Flexural Rigidity of the Superstructure", Proc 6th Int Conf Soil Mech Fdn Eng, vol 2, pp 197-201

SVED, G and KWOK, H L (1963)* "The effect of non-linear foundation settlement on the distribution of bending moments in a building frame", Proc 4th Australia-New Zealand Conf Soil Mech Fdn Eng, pp 18-22

TERZAGHI, K (1923)* "Die Berechnung der Durchlassigkeitsziffer des Tones auf dem Verlauf der hydrodynamischen Spannungserscheinungen", Sitz Akad Wien Math-naturw Kl Abt 11a, 132, pp 105-124

TERZAGHI, K (1943) "Theoretical Soil Mechanics", J Wiley, New York, 13th printing, 1965

TIMOSHENKO, S P and GOODIER, J N (1970) "Theory of Elasticity", McGraw-Hill, 3rd ed

UESHITA, K and MEYERHOF, G G (1968)* "Surface Displacement of an Elastic Layer under uniformly distributed loads", Highway Res Record No 228, pp 1-10

VAHABI, H (1978) "An experimental investigation into soil-structure interaction", University of Bath, School of Architecture and Building Engineering, Project Report

VALLIAPPAN, S (1975) "Application of Finite Element Method to Soil Deformation", Proc Gen Sess Symp Recent Dev Soil Mech, Univ NSW (1975), pp 113-142

VIAGGIANI, C; DAVIS, E H and POULOS, H G (1970) Discussion, Jnl Soil Mech Fdn Div Am Soc Civ Engrs, vol 96, pp 331-336

WOOD, L A (1972) "Some Aspects of Soil-Structure Interaction", PhD thesis, Univ of Bristol, (2 vols)

WOOD, L A and LARNACH, W J (1975) "The effects of soil-structure interaction on raft foundations", in "Settlement of Structures", pp 460-470, Pentech Press, London

YOKOO, YOSHITSURA, KUNIO YAMAGATA and HIROAKI NAGAOKA (1971a)* "Finite Element Method applied to Biot's Consolidation Theorem", Soils Found, vol 11, No 1, pp 29-46

YOKOO, YOSHITSURA, KUNIO YAMAGATA and HIROAKI NAGAOKA (1971b)* "Finite Element Analysis of Consolidation following Undrained Deformation", Soils Found, vol 11, No 4, pp 37-58

ZIENKIEWICZ, O C and NAYLOR, D J (1972) "The Adaptation of critical state soil mechanics theory for use in finite elements", in "Stress-strain behaviour of soils", Parry, R H G ed, G T Foulis and Co (London), pp 537-547

ZIENKIEWICZ, O C (1977) "The Finite Element Method", McGraw-Hill (London), 3rd ed

ADDENDUM

CHEUNG, Y K and ZIENKIEWICZ, O C (1965)* "Plates and tanks on elastic foundation: an application of finite element method", Int J.Solids Struct, vol 1, pp 451-461

MURRAY, R T (1972) "Computer program for the one-dimensional analysis of the rate of consolidation of multi-layered soils", Transport and Road Res Lab report LR443

MURRAY, R T (1974) "Two-dimensional analysis of settlement by computer program", Transport and Road Res Lab report 617

MURRAY, R T and SYMONS, I F (1974) "Embankments on soft foundations: settlement and stability study at Tickton in Yorkshire", Transport and Road Res Lab report 643

SMALL, J C; BOOKER, J R and DAVIS, E H (1975) "Elasto-plastic consolidation of soil", Univ of Sydney, School of Civil Eng, Res Report No R268

APPENDIX A - Program specification: FDTIM

The program FDTIM is written in FORTRAN IV using double precision arithmetic. It models the plane strain consolidation of a soil layer under various types of surface loading. The theory and structure of the program are detailed in Chapters 2 and 3 of this thesis. We here summarize the roles of the various subroutines (in the order in which they are called in the main program), and give instructions for data input.

Main Program

The main program is dimensioned for a mesh with a maximum of 70 vertical columns (including end ones) and 40 horizontal rows (including surface and base). Up to 9 loading times and 9 loaded footings are allowed. If these dimensions are to be changed, the DATA statement (giving these sizes as MX, MZ, MLOAD and MFOOT respectively) must be adjusted, as well as the initial DIMENSION statement. No changes need be made to the subroutines.

Subroutine INPUT

This subroutine reads all data input, including loads, in the formats shown below:

Number of Cards	Format	Variables
1	12I5,I10,D10.3	NX,NZ,NFOOT,NLOAD,NDTMX,LAYER,INTER, ISTRE,ISUPN,NDTIM,NTMAG,NTFAC, NTEND, ZSCAL.
1	8I5	(ICOLPT(I), I = 1,8)
as reqd	8D10.3	(XLOC(I),I = 1,NX)
NFOOT	4I5	I1(I),IM(I),I2(I),ITYPE(I) for I=1, NFOOT
NZ[1]	4D10.3,I5	ZLOC(J),PERMX(J),PERMZ(J),PR(J), IFACE(J) for J = 1, NZ
NZ[1]	I5,5X,3D10.3	LMTYP(J),YMINIT(J),CC(J),PHI(J) for J = 1, NZ
1	8D10.3	AP,BP,WDENS,GRAVY,WIDTH,DRMAX,RFACT, EPS

CONT'D .....

..... CONT'D

Number of Cards	Format	Variables
NZ^2	2D10.3	SXINIT(J),SZINIT(J) for J=1,NZ
1	5I1	(NBDRY(I), I = 1,5)
NLOAD { 1	8D10.3	(QEXT(NSET,ILOAD),NSET=1,8) } ILOAD=1, NLOAD
NLOAD { as reqd	I1,I4,8I5	$LCARD^4$,LDTIM(ILOAD),(ILDPT (NSET),NSET=1,8) } ILOAD=1, NLOAD
as reqd[3]	8D10.3	(EI(IBAY),IBAY=1,NFOOT-1)

Notes

1. For the input of arrays PERMX, PERMZ, PR, LMTYP, YMINIT, CC and PHI, which hold soil properties at the depths of the NZ mesh rows, the following default convention applies:

   for J = 2, ..., NZ - 1, if the space for eg. CC(J) is left blank, its value defaults to that for the row directly above, ie. CC(J - 1). For the soil properties at boundaries between different soil layers (indicated by IFACE), the program automatically averages the values from the rows immediately above and below.

2. Cards for SXINIT and SZINIT should be omitted if there are no initial stresses (shown by ISTRE ≠ 0)

3. Cards for EI should be omitted if there is no beam element connecting the footings (hereafter referred to as a non-interaction program and indicated by INTER = 0).

4. LCARD = 1 to denote the last card of a particular type.

   The meanings of the input variables are now given:

Input data: Simple Variables

Name	Description	Default
NX	No. of columns in mesh	none
NZ	No. of rows in mesh	none

CONT'D .....

..... CONT'D

Name	Description	Default
NFOOT	No of loaded footings	1
NLOAD	No of load times	1
NDTMX	Max. size of timestep	99999
LAYER	= 0 for infinite-depth Boussinesq stresses = 1 for finite-depth Filon stresses	0
INTER	= 0 for independent footings (non-interaction problem) = 1 for footings linked by beam	0
ISTRE	= 0 if initial stress state of the soil is prescribed = 1 if no initial stress state given	0
ISUPN	= 0 for full output at all mesh points =-1 for output only at columns listed in ICOLPT =-2 for output only at surface, for columns listed in ICOLPT	0
NDTIM	initial timestep	2
NTMAG	No of steps after which NDTIM is increased	5
NTFAC	Factor by which NDTIM is increased	10
NTEND	Time at which timestepping ends	none
ZSCAL	Multiplying factor applied to depths as input for ZLOC. (May be used to analyse the same problem for various depths of soil layer)	1.0
AP	Skempton's triaxial pore-pressure coefficients	0.0
BP		1.0
WDENS	Water density	9.81
GRAVY	Gravity force	0.00981
WIDTH	Out-of-plane thickness	1.0
DRMAX	Acceptable error in footing displacements to stop iteration (used in interaction problems only)	10^{-4}
RFACT	Adjusting factor R_f in hyperbolic stress-strain law	0.8
EPS	Parameter m is contact stress distribution for rigid footings	2.0

Input data: arrays

Name & Max. Dimensions	Description
ICOLPT(8)	Columns at which output is required (used if ISUPN < 0)
	Description of I'th entry
XLOC(MX)	x-coord of I'th column in mesh
I1(MFOOT)	No of mesh column below left hand edge of I'th footing
IM(MFOOT)	No of mesh column below mid-point of I'th footing
I2(MFOOT)	No of mesh column below right hand edge of I'th footing.
ITYPE(MFOOT)	Loading type at I'th footing (=1,2,3,4;see Fig.A1)
	Description of J'th entry
ZLOC(MZ)	z-coord (=depth) of J'th row in mesh
PERMX(MZ)	Horizontal permeability at J'th row in mesh
PERMZ(MZ)	Vertical permeability at J'th row in mesh
PR(MZ)	Poisson's ratio at J'th row in mesh
IFACE(MZ)	1 if J'th row is a boundary between different soil layers; 0 otherwise.
LMTYP(MZ)	Soil type at J'th row (2=lin.elastic,3=hyperbolic model, 4=bilinear)
YMINIT(MZ)	Initial Young's modulus at J'th row in mesh
CC(MZ)	For hyperbolic model,denotes cohesion c' at J'th row for 'bilinear' model, denotes factor c: $E_{unload} = cE_{load}$ at J'th row
PHI(MZ)	For hyperbolic model, denotes angle of friction Ø' at J'th row.
SXINIT(MZ)	Initial effective stress σ'_x at J'th row
SZINIT(MZ)	Initial effective stress σ'_z at J'th row

CONT'D .....

..... CONT'D

Name & Max. Dimensions	Description
NBDRY (5)	Boundary conditions. Code: 0=permeable, 1=impermeable. NBDRY (1) for base NBDRY (2) for left hand end of mesh NBDRY (3) for right hand end of mesh NBDRY (4) for surface beneath footings NBDRY (5) for surface away from footings
QEXT(8,MLOAD) ILDPT(8)	QEXT(NSET,I) = size of load applied at I'th load time to load point given by ILDPT(NSET)
LDTIM(MLOAD)	LDTIM(I) = time of I'th loading
EI(MFOOT-1)	EI(I) = stiffness EI of beam at I'th bay.

Timestepping Scheme

This has been made very flexible, as the most efficient scheme is highly problem-dependent. The initial timestep NDTIM is increased by a factor of NTFAC after every NTMAG steps, provided it does not become larger than an upper limit NDTMX. Each time a new set of loads is applied, the timestep reverts to the initial value NDTIM. The program run finishes when the end-time NTEND is reached. The timestepping scheme operates in integer arithmetic, so that the smallest possible value of NDTIM is 1. Should a smaller timestep be needed, the same effect is achieved by reducing the permeabilities.

Loading Scheme

The times at which loads are to be applied are held in LDTIM. At each load time, loads may be applied at any load point. Load points are defined as follows:

- for non-interaction problems, there are NFOOT load points, one at each footing.
- for interaction problems, with a portal structure of NFOOT-1 bays linking the footings, the load points are on the connecting beam at the ends and mid points of each bay (a total of 2*NFOOT-1 load points)

The load points are numbered consecutively from left to right for the purposes of input using ILDPT.

Note that only point loads, not pressures, are used. The program converts from point load to average pressure on a footing using GRAVY and WIDTH.

Symmetrical Problems

For a non-interaction problem with symmetry about the centreline, only half of the problem need be modelled. The centreline must be placed at the left hand side of the mesh, ie. down the first column, which should be an impermeable boundary. When Il(1) = 1, the program assumes each loaded footing to have a mirror image equivalent on the other side of the centreline. Note that if the first footing is rigid, we set IM(1) = 1 also.

Subroutine STIFF

Variables in	Variables out
NFOOT, EI, IM,XLOC	GRHS

Used in interaction problems only. Forms the beam flexibility matrix GRHS used in subroutine STRUCT to convert beam loads to footing loads. Theory given in § 3.4.

Subroutine STRUCT

Variables in	Variables out
QEXT, DISPF, ILOAD, NFOOT, GRHS	Q, ROTN, DISPL

Used in interaction problems only. Takes the external beam loads specified in QEXT for loading ILOAD and the current footing displacements DISPF, and using the flexibility matrix GRHS determines the footing loads Q and the beam load point rotations ROTN and displacements DISPL. Theory in § 3.4.

Subroutine FILON

Variables in	Variables out
DSX, DSZ, DTXZ, XL, XR, DPRESS, NX, NZ, XLOC, XLOC, ITYP, EPS, IL, IR	DSX, DSZ, DTXZ

Adds to the mesh point total stress increment arrays DSX, DSZ, DTXZ the increments resulting from an average pressure DPRESS on a footing of type

ITYP lying between columns IL and IR with x-coords XL and XR. Uses the finite-layer Filon stress distribution theory, see § 2.2.

Subroutine BOUSS

Same variables and action as FILON, but uses the infinite-depth Boussinesq stress distribution theory (see § 2.2).

Subroutine PRINC

Variables in	Variables out
CX, CZ, CT	S1, S3

Converts cartesian coordinate stresses σ_x, σ_z, τ_{xz} to principal stresses σ_1, σ_3.

Subroutine OUTPUT

Outputs structural displacements DISPL and rotations ROTN (interaction problems only), footing displacements DISPF, and mesh point excess stresses SX, SZ, TXZ, displacements DISPZ and excess pore pressures U. Also outputs time NTIME, number of iterations NITER (interaction problems only), and number of steps taken since last loading NSTEP.' Output may be limited using ISUPN and ICOLPT, as described under INPUT.

Subroutine ADIDE

Variables in	Variables out
NX, NZ, NDTIM, XLOC, ZLOC, PERMX, PERMZ, YM, PR, WDENS, U, NBDRY, NFOOT, I1, I2, IFACE.	U

Advances mesh point excess pore pressure distribution U by one timestep, using the ADI finite difference method (see §3.3). There is a satellite subroutine TRIDG to solve the tridiagonal matrix equations involved.

Subroutine YMHYP

Variables in	Variables out
J, YM1, RFACT, PHI, CC, S1, S3, DS	YM1

Converts the initial Young's modulus YM1 to that appropriate to the current stress state (with principal stresses S1, S3 acting) as determined by the hyperbolic stress-strain model of § 5.3. If the vertical strain is expansive (shown by DS<O) then YM1 is unchanged. Used when LMTYP(J) = 3

Subroutine YMLIN

Variables in

J, YM1, CC, DS

Variables out

YM1

Same role as YMHYP, but using a 'bilinear' load/unload model. If an unload is occurring (shown by DS<0), then YM1 is changed to YM1*CC(J), Used when LMTYP(J) = 4.

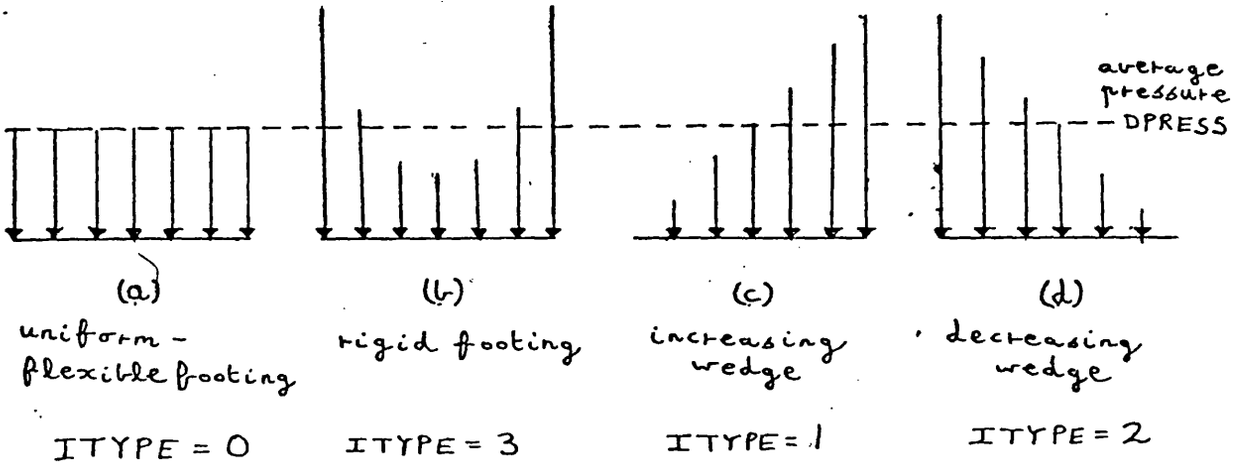


Fig.A1 LOADING TYPES

Program Listing

Program FDTIM

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IHEAD(17),Q(9),I1(9),IM(9),I2(9),DISPF(9),DISPL(17),
     1DFOLD(9),QEXT(17,9),NBDRY(5),EI(9),GRHS(34,34),RQTN(17),
     2DISPZ(70,40),SZ(40,70),SX(40,70),TXZ(40,70),DSZ(40,70),
     3DSX(40,70),DTXZ(40,70),STRAIN(70,40),U(70,40),V(70,40),
     4UOLD(70,40),YH(70,40),XLOC(70),ZLOC(40),YHINIT(40),
     5CC(40),PHI(40),LMTYP(40),PR(40),LDTIM(9),PERMX(40),PERMZ(40),
     6A(70),B(70),C(70),D(70),X(70),ST1(70),ST2(70),ST3(70),
     7GSTIF(34,34),VLHS(34),VRHS(34),ITYPE(9),ICOLPT(8)
     8,SXINIT(40),SZINIT(40),IFACE(40),QOLD(9)
C
C
C...FIX MAX. DIMENSIONS - MUST TALLY WITH DIM. STATEMENT ABOVE
      DATA MX/70/, MZ/40/, MLOAD/9/, MFOOT/9/
      MLDPT = 2*MFOOT - 1
      MSIZE = 2*MLDPT
C
C...READ AND WRITE HEADING
      IHEAD(1) = 63
      WRITE (6,597)
    1 READ (5,500) LCARD,(IHEAD(I),I=2,17)
  500 FORMAT (I1,16A4)
      WRITE (6,598) (IHEAD(I),I=2,17)
      IF (LCARD.EQ.0) GO TO 1
      WRITE (6,599)
  597 FORMAT(1H0,//' PROGRAM FDTIM'//1X,85(1H*),/1X,1H*,83X,1H*)
  598 FORMAT(1X,1H*,2X,16A4,17X,1H*)
  599 FORMAT(1X,1H*,83X,1H*,/1X,85(1H*))
C
      DO 3 J=1,MZ
      SXINIT(J) = 0.0D0
      SZINIT(J) = 0.0D0
      DO 3 I=1,MX
      UOLD(I,J) = 0.0D0
      V(I,J) = 0.0D0
```

```
      U(I,J) = 0.0D0
      SX(J,I) = 0.0D0
      SZ(J,I) = 0.0D0
      TXZ(J,I) = 0.0D0
    3 CONTINUE
C
      CALL INPUT(MX,MZ,MMAX,MFOOT,MLOAD,MLDPT,NX,NZ,NFOOT,NLOAD,NDTMX
     1,NDTIM,NTMAG,NTFAC,NTEND,LAYER,INTER,ISUPN,I1,IM,I2,XLOC,ZLOC,
     2YMINIT,PR,CC,PHI,PERMX,PERMZ,AP,BP,WDENS,GRAVY,WIDTH,DRMAX,
     3RFACT,EPS,NBDRY,LDTIM,QEXT,LMTYP,EI,SXINIT,SZINIT,
     4ITYPE,ICOLPT,IFACE)
C
C...FIND PLANE-STRAIN PORE-PRESSURE COEFFT APS
      NLDPT = NFOOT
      IF (INTER.NE.0) NLDPT = 2*NFOOT - 1
      D3 = DSQRT(3.0D0)
      APS = (1. - 1./D3 + AP*D3)/2.
      IF (INTER.EQ.1) CALL STIFF(MX,NFOOT,MSIZE,NFOOT,EI,IM,
     1XLOC,GRHS,GSTIF)
C...INITIALIZE VECTORS  AND MATRICES
      NX1 = NX - 1
      NZ1 = NZ - 1
      NPTIM = 0
      NTIME = 0
      ILOAD = 0
      NTINIT = NDTIM
C...USE ISYMM=1 TO DENOTE MODEL SYMMETRIC ABOUT L.H.S.
      ISYMM = 0
      IF (I1(1).EQ.1) ISYMM = 1
      DO 5 J=1,NZ
      YMJ = YMINIT(J)
      DO 5 I=1,NX
      DISPZ(I,J) = 0.0D0
      DSZ(J,I) = 0.0D0
      DSX(J,I) = 0.0D0
      DTXZ(J,I) = 0.0D0
      STRAIN(I,J) = 0.0D0
      YM1 = YMJ
      IF (LMTYP(J).EQ.3) CALL YMMYP(MZ,J,YM1,RFACT,PHI,CC,
     1SZINIT(J),SXINIT(J),1.0D0)
      YM(I,J) = YM1
    5 CONTINUE
      DO 6 I=1,NFOOT
      GOLD(I) = 0.0D0
      G(I) = 0.0D0
    6 DISPF(I) = 0.0D0
      DO 7 I=1,NLDPT
      DISPL(I) = 0.0D0
    7 ROTN(I) = 0.0D0
      DREL = 0.0D0
C
C
C...ADD NEXT SET OF LOADS
   10 ILOAD = ILOAD + 1
      NITER = 0
      NSTEP = 0
      NDTIM = NTINIT
      DO 13 I=1,NLDPT
      QI = QEXT(I,ILOAD)
      IF (QI.EQ.0.0D0) GO TO 13
```

```
      WRITE (6,600) ILOAD,QI,I
  600 FORMAT(1H ,'LOAD',I3,' OF ',D10.3,' AT LOAD-PT',I3)
   13 CONTINUE
C
C... FIND FOOTING LOADS
   14 IF (INTER.EQ.1) GO TO 16
      DO 18 I=1,NLDPT
   18 Q(I) = QEXT(I,ILOAD)
      GO TO 19
   16 CALL STRUCT(QEXT,Q,DISPF,ILOAD,MFOOT,MLDPT,NFOOT,
     1NLOAD,GRHS,MSIZE,ROTN,DISPL,VLHS,VRHS)
C
      IF (NTIME.EQ.LDTIM(ILOAD).AND.NITER.EQ.0)
     1WRITE (6,601) (Q(J),J,J=1,NFOOT)
  601 FORMAT(1H ,'GIVES REACTIONS:',5X,5(F7.3,' AT FTNG ',I2,5X))
   19 DO 21 I=1,NFOOT
C...CONVERT LOAD IN KG TO PRESSURE IN KN/M2 & FIND STRESS INCS
      IL = I1(I)
      IR = I2(I)
      XL = XLOC(IL)
      XR = XLOC(IR)
      ITYP = ITYPE(I)
      AREA = WIDTH*(XR-XL)
      DPRESS = (Q(I) - QOLD(I))*GRAVY/AREA
      IF (LAYER.EQ.0) GO TO 12
      CALL FILON(DSZ,DSX,DTXZ,XL,XR,DPRESS,NZ,NX,XLOC,ZLOC,
     1NX,NZ,ITYP,EPS,IL,IR)
      GO TO 20
   12 CONTINUE
      CALL BOUSS(DSZ,DSX,DTXZ,XL,XR,DPRESS,NZ,NX,XLOC,ZLOC,
     1NX,NZ,ITYP,EPS,IL,IR)
   20 IF (ISYMM.EQ.0) GO TO 21
C...FOR SYMMETRIC PROBLEMS, ADD STRESSES FROM MIRROR-IMAGE LOAD
      X1 = XLOC(1)
      IL1 = 2 - IR
      IR1 = 2 - IL
      XR1 = 2.*X1 - XL
      XL1 = 2.*X1 - XR
      IF (ITYP.EQ.1.OR.ITYP.EQ.2) ITYP = 3 - ITYP
      IF (LAYER.EQ.0) GO TO 11
      CALL FILON(DSZ,DSX,DTXZ,XL1,XR1,DPRESS,NZ,NX,XLOC,ZLOC,
     1NX,NZ,ITYP,EPS,IL1,IR1)
      GO TO 21
   11 CONTINUE
      CALL BOUSS(DSZ,DSX,DTXZ,XL1,XR1,DPRESS,NZ,NX,XLOC,ZLOC,
     1NX,NZ,ITYP,EPS,IL1,IR1)
   21 CONTINUE
      NITER = NITER + 1
C
C - FIND UNDRAINED PORE-PRESSURE INCREASES & NEW STRESSES
      DO 15 I=1,NX
      DO 15 J=1,NZ
      CX = SX(J,I) + SXINIT(J)
      CZ = SZ(J,I) + SZINIT(J)
      CT = TXZ(J,I)
      CALL PRINC(CX,CZ,CT,S1OLD,S3OLD)
      SX(J,I) = SX(J,I) + DSX(J,I)
      SZ(J,I) = SZ(J,I) + DSZ(J,I)
      TXZ(J,I) = TXZ(J,I) + DTXZ(J,I)
      CX = SX(J,I) + SXINIT(J)
```

```
      CZ = SZ(J,I) + SZINIT(J)
      CT = TXZ(J,I)
      CALL PRINC(CX,CZ,CT,S1NEW,S3NEW)
      DS1 = S1NEW - S1OLD
      DS3 = S3NEW - S3OLD
      DU = BP*(DS3 + APS*(DS1-DS3))
      U(I,J) = U(I,J) + DU
   15 CONTINUE
C
C...NOW FIND IMMEDIATE STRAINS
      DO 71 J=1,NZ
      PRJ = PR(J)
C...FIND FACTOR TO CONVERT DRAINED TO UNDRAINED MODULI
      COEFF = 1. - AP*(1.-2.*PRJ)
      IF (DABS(COEFF).LT.1.0D-6) GO TO 400
      DO 71 I=1,NX
      YMIJ = YM(I,J)/COEFF
      DS = 0.75*(DSZ(J,I) - DSX(J,I))
   71 STRAIN(I,J) = STRAIN(I,J) + DS/YMIJ
C...FIND DISPLACEMENTS AND ZERO STRAIN,DSZ,ETC.
      DO 25 I=1,NX
      DDISP = 0.0D0
      DO 25 JR=1,NZ1
      J = NZ - JR
      H = ZLOC(J+1) - ZLOC(J)
      DDISP = DDISP + 0.5*H*(STRAIN(I,J) + STRAIN(I,J+1))
      DISPZ(I,J) = DISPZ(I,J) + DDISP
   25 CONTINUE
      DO 30 I=1,NX
      DO 30 J=1,NZ
      STRAIN(I,J) = 0.0D0
      DSZ(J,I) = 0.0D0
      DSX(J,I) = 0.0D0
      DTXZ(J,I) = 0.0D0
   30 CONTINUE
   29 CONTINUE
C...IF RELATIVE SETTLEMENT TOO BIG, ITERATE AGAIN
      DO 26 IFOOT=1,NFOOT
      INFT = IN(IFOOT)
      QOLD(IFOOT) = Q(IFOOT)
      DFOLD(IFOOT) = DISPF(IFOOT)
   26 DISPF(IFOOT) = DISPZ(INFT,1)
C...FOR UNLINKED FOOTINGS, NO ITERATION NECESSARY
      IF (INTER.EQ.0) GO TO 24
C...ALWAYS ITERATE AT LEAST ONCE
      IF (NITER.EQ.0) GO TO 14
      DREL = 0.0D0
      DO 27 I=1,NFOOT
      DIFF = DABS(DISPF(I) - DFOLD(I))
      IF (DIFF.GT.DREL) DREL = DIFF
   27 CONTINUE
      IF (DREL.GT.DRMAX) GO TO 14
   24 CONTINUE
C...ZERO QOLD
      DO 23 I=1,NFOOT
   23 QOLD(I) = 0.0D0
      CALL OUTPUT(NX,NZ,SX,SZ,DISPZ,U,DISPF,NFOOT,NTIME,NITER,
     1NX,NZ,NFOOT,NSTEP,ROTN,NLDPT,INTER,DISPL,Q,ISUPN,TXZ,ICOLPT)
C...RETURN IF FURTHER LOAD TO BE ADDED
      IF (ILOAD.EQ.NLOAD) GO TO 28
```

```
      IF (NTIME.EQ.LDTIM(ILOAD+1)) GO TO 10
   28 CONTINUE
C
C...PUT OLD U INTO UOLD, APPLYING BDRY CONDNS IF NECESSARY
      IF (NITER.GT.0) GO TO 36
      DO 35 J=1,NZ
      DO 35 I=1,NX
   35 UOLD(I,J) = U(I,J)
      GO TO 37
   36 CONTINUE
      IF (NBDRY(2).EQ.0) GO TO 61
      DO 60 J=2,NZ1
      V(1,J) = U(1,J)
   60 UOLD(1,J) = U(1,J)
      GO TO 63
   61 DO 62 J=1,NZ
      UOLD(1,J) = U(1,J)
   62 U(1,J) = 0.0D0
   63 IF (NBDRY(3).EQ.0) GO TO 57
      DO 56 J=2,NZ1
      V(NX,J) = U(NX,J)
   56 UOLD(NX,J) = U(NX,J)
      GO TO 59
   57 DO 58 J=1,NZ
      UOLD(NX,J) = U(NX,J)
   58 U(NX,J) = 0.0D0
   59 CONTINUE
      IF (NBDRY(1).EQ.0) GO TO 65
      DO 64 I=1,NX
      V(I,NZ) = U(I,NZ)
   64 UOLD(I,NZ) = U(I,NZ)
      GO TO 67
   65 DO 66 I=1,NX
      UOLD(I,NZ) = U(I,NZ)
   66 U(I,NZ) = 0.0D0
   67 IFOOT = 1
      NBC = NBDRY(5)
      IF (I1(1).EQ.1) NBC = NBDRY(4)
      DO 70 I=1,NX
      IF (I.EQ.I1(IFOOT)+1) NBC = NBDRY(4)
      IF (I.NE.I2(IFOOT)) GO TO 68
      IF (I.EQ.NX) GO TO 68
      NBC = NBDRY(5)
      IF (IFOOT.LT.NFOOT) IFOOT = IFOOT + 1
   68 CONTINUE
      IF (NBC.EQ.0) GO TO 69
      V(I,1) = U(I,1)
      UOLD(I,1) = U(I,1)
      GO TO 70
   69 UOLD(I,1) = U(I,1)
      U(I,1) = 0.0D0
   70 CONTINUE
      DO 75 I=2,NX1
      DO 75 J=2,NZ1
   75 UOLD(I,J) = U(I,J)
   37 NITER = 0
C
C...NOW LET PORE-WATER DIFFUSE OVER ONE TIMESTEP
      IF (MOD(NSTEP,NTFAC).EQ.0.AND.NSTEP.GT.0.AND.NDTIM.LT.NDTMX)
     1NDTIM = NDTIM*NTFAC
```

```
      NTIME = NTIME + NDTIN
      NSTEP = NSTEP + 1
      CALL ADIDE(MMAX,NX,NZ,NDTIN,XLOC,ZLOC,PERMX,PERMZ,YM,PR,WDENS,
     1U,V,NBDRY,NFOOT,I1,I2,MX,MZ,NFOOT,A,B,C,D,X,ST1,ST2,ST3,IFACE)
C
C...FIND STRAINS AND DISPLACEMENTS
      DO 40 J=1,NZ
      YMJ = YMINIT(J)
      PRJ = PR(J)
      DO 40 I=1,NX
      DU = U(I,J) - UOLD(I,J)
      DS = (1.+PRJ) * (2.*PRJ-1.) * DU
      CX = SX(J,I) + SXINIT(J)
      CZ = SZ(J,I) + SZINIT(J)
      CT = TXZ(J,I)
      CALL PRINC(CX,CZ,CT,S1,S3)
      S1 = S1 - U(I,J)
      S3 = S3 - U(I,J)
      YM1 = YMJ
      IF (LMTYP(J).EQ.3) CALL YMMYP(MZ,J,YM1,RFACT,PHI,CC,S1,S3,DS)
      IF (LMTYP(J).EQ.4) CALL YMLIN(MZ,J,YM1,CC,DS)
      YM(I,J) = YM1
   40 STRAIN(I,J) = DS/YM(I,J)
      DO 55 I=1,NX
      DDISP = 0.0D0
      DO 55 JR=1,NZ1
      J = NZ - JR
      H = ZLOC(J+1) - ZLOC(J)
      DDISP = DDISP + 0.5*H*(STRAIN(I,J) + STRAIN(I,J+1))
      DISPZ(I,J) = DISPZ(I,J) + DDISP
   55 CONTINUE
      IF (NTIME.LE.NTEND) GO TO 29
      STOP
  400 WRITE (6,700) J
  700 FORMAT('0','ERROR: CANNOT EVALUATE UNDRAINED MOD. IN LEVEL'
     1,I5/1X,'PORE-PRESSURE COEFFT. A MUST BE CHANGED.')
      STOP
      END
C
C
C...SUBROUTINES
C
C
      SUBROUTINE BOUSS(DSZ,DSX,DTXZ,X1,X2,Q,NZ,NX,XLOC,ZLOC,
     1MX,MZ,ITYPE,EPS,I1,I2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DSZ(MZ,MX),DSX(MZ,MX),DTXZ(MZ,MX),XLOC(MX),ZLOC(MZ)
      PI = 3.14159
      QEPS = 1. - 1./((0.609522*EPS-0.0166826)*EPS+0.345663)
      DEPTH = ZLOC(NZ) - ZLOC(1)
      XM = (X1 + X2)/2.
      B = XM - X1
      IF (ITYPE.NE.3) GO TO 3
C...MID-POINT FOR RIGID FOOTING IN SYMM. PROBLEM MAY DIFFER
      IF (X1.EQ.XLOC(1)) XM = X1
      IF (X2.EQ.XLOC(1)) XM = X2
      B = DMAX1(X2-XM,XM-X1)
    3 CONTINUE
C...MAKE H SMALL COMPARED WITH MESH SIZE
      ID = I2 - I1 + 2
```

```
      DH = (X2-X1)/DFLOAT(5*ID)
      QH = Q*DH
      DO 12 JV = 2,NZ
      Z = ZLOC(JV) - ZLOC(1)
      DO 10 IV=1,NX
      SIGX = 0.0D0
      SIGZ = 0.0D0
      SIGT = 0.0D0
      XLOCB = X1 + DH/2.
    5 CONTINUE
      X = XLOCB - XLOC(IV)
      DB = (XH - XLOCB)/B
      QB = QH
      IF (ITYPE.EQ.1) QB = QB*(1.-DB)
      IF (ITYPE.EQ.2) QB = QB*(1.+DB)
      IF (ITYPE.EQ.3) QB = QB*QEPS/((1.-DABS(DB)**EPS)**(1./EPS))
      R2 = (X*X + Z*Z)**2
      ZR2 = Z/R2
      DSIGX = X*X*ZR2
      DSIGZ = Z*Z*ZR2
      DSIGT = X*Z*ZR2
      COEFF = 2.*QB/PI
      SIGX = SIGX + DSIGX*COEFF
      SIGZ = SIGZ + DSIGZ*COEFF
      SIGT = SIGT + DSIGT*COEFF
      XLOCB = XLOCB + DH
      IF (XLOCB.LT.X2) GO TO 5
      DSX(JV,IV) = DSX(JV,IV) + SIGX
      DSZ(JV,IV) = DSZ(JV,IV) + SIGZ
      DTXZ(JV,IV) = DTXZ(JV,IV) + SIGT
   10 CONTINUE
   12 CONTINUE
C
C...STRESSES ON SURFACE
      DO 20 IV=1,NX
      X = XLOC(IV)
      IF (X.GT.X2) RETURN
      IF (X.LT.X1) GO TO 20
      QB = Q
      DB = (XH-X)/B
      COEFF = 1.0D0
C...ONLY HALF THE CONTACT STRESS AT THE FOOTING EDGES
      IF (IV.EQ.I1.OR.IV.EQ.I2) COEFF = 0.5
      IF (ITYPE.NE.3) GO TO 15
C...ADJUST X TO AVOID INFINITE STRESSES AT RIGID FTG EDGES
      IF (COEFF.EQ.1.0D0) GO TO 14
      IF (DB.GT.0.0D0) X = (X+XLOC(IV+1))/2.
      IF (DB.LT.0.0D0) X = (XLOC(IV-1)+X)/2.
      DB = (XH-X)/B
   14 QB = QB*QEPS/((1.-DABS(DB)**EPS)**(1./EPS))
   15 CONTINUE
      IF (ITYPE.EQ.1) QB = QB*(1.-DB)
      IF (ITYPE.EQ.2) QB = QB*(1.+DB)
      DSX(1,IV) = DSX(1,IV) + COEFF*QB
      DSZ(1,IV) = DSZ(1,IV) + COEFF*QB
   20 CONTINUE
      RETURN
      END
C
C
```

```
      SUBROUTINE ADIDE(MMAX,NX,NZ,NDTIM,XLOC,ZLOC,PERMX,PERMZ,YM,PR,
     1WDENS,U,V,NBDRY,NFOOT,I1,I2,MX,MZ,NFOOT,A,B,C,D,X,ST1,ST2,ST3,
     2IFACE)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XLOC(MX),ZLOC(MZ),PERMX(MZ),PERMZ(MZ),YM(MX,MZ),PR(MZ),
     1U(MX,MZ),V(MX,MZ),ST1(MMAX),ST2(MMAX),ST3(MMAX),A(MMAX),B(MMAX),
     2C(MMAX),D(MMAX),X(MMAX),NBDRY(5),I1(NFOOT),I2(NFOOT),IFACE(MZ)
C
C.....ADVANCES PORE-PRESSURE DISTRIBUTION U(NX,NZ) BY ONE TIMESTEP
C.....BY A.D.I. SOLN OF 2D DIFFUSION EQN.
C
      NX1 = NX-1
      NX2 = NX-2
      NZ1 = NZ-1
      NZ2 = NZ-2
      DTIME = DFLOAT(NDTIM)/2.
      NOGO = 0
C
C..... FIRST EQUATION
C
      NBC1 = NBDRY(1)
      NBC2 = NBDRY(2)
      NBC3 = NBDRY(3)
      ILIM1 = 2 - NBC2
      ILIM2 = NX1 + NBC3
      DO 10 J=2,NZ1
      DZ1 = ZLOC(J) - ZLOC(J-1)
      DZ2 = ZLOC(J+1) - ZLOC(J)
C...ALLOW FOR LEVEL J BEING INTERFACE BETWEEN SOIL LAYERS
      PZ1 = PERMZ(J)
      PZ2 = PZ1
      IF (IFACE(J).EQ.0) GO TO 3
      PZ1 = PERMZ(J-1)
      PZ2 = PERMZ(J+1)
    3 CONTINUE
      CP = 2.*DTIME*PZ1/(WDENS*DZ1)
      CQ = 2.*DTIME*PZ2/(WDENS*DZ2)
      Z1 = 2.*DZ2/(DZ1+DZ2)
      Z2=2.-Z1
      CT = 2.*(1.+PR(J))*(1.-2.*PR(J))
      IVEC = 0
      DO 5 I=ILIM1,ILIM2
      IVEC = IVEC + 1
      CJM1 = CT/YM(I,J-1)
      CJ   = CT/YM(I,J)
      CJP1 = CT/YM(I,J+1)
      T = CJM1*DZ1 + CJP1*DZ2
      P = CP/T
      Q = CQ/T
      B(IVEC) = P*Z1*U(I,J-1) + (1.-P-Q)*U(I,J) + Q*Z2*U(I,J+1)
      IF (I.EQ.1) DX1 = XLOC(2) - XLOC(1)
      IF (I.NE.1) DX1 = XLOC(I) - XLOC(I-1)
      IF (I.EQ.NX) DX2 = XLOC(NX) - XLOC(NX1)
      IF (I.NE.NX) DX2 = XLOC(I+1) - XLOC(I)
      X1 = 2.*DX2/(DX1+DX2)
      X2 = 2. - X1
      R = PERMX(J)*DTIME/(WDENS*CJ*DX1*DX2)
      A(IVEC) = -R*X1
      B(IVEC) = 1. + 2.*R
```

```
      C(IVEC) = -R*X2
    5 CONTINUE
C..... SIDE BDRY CONDITIONS
      IF (NBC2.EQ.1) C(1) = C(1) + A(1)
      IF (NBC3.EQ.1) A(IVEC) = A(IVEC) + C(IVEC)
      CALL TRIDG(A,B,C,D,X,IVEC,NOGO,ST1,ST2,ST3)
      IVEC = 0
      DO 8 I=ILIM1,ILIM2
      IVEC = IVEC + 1
    8 V(I,J) = X(IVEC)
   10 CONTINUE
C...SOLVE ALONG HORIZ. IMP. BDRIES
C...FIRST ALONG BASE
      IF (NBC1.EQ.0) GO TO 44
      DZ = ZLOC(NZ) - ZLOC(NZ1)
      CP = DTIME*PERMZ(NZ1)/(WDENS*DZ*DZ)
      CT = 2.*(1.+PR(NZ))*(1.-2.*PR(NZ))
      IVEC = 0
      DO 42 I=ILIM1,ILIM2
      IVEC = IVEC + 1
      CJM1 = CT/YM(I,NZ1)
      CJ = CT/YM(I,NZ)
      P = CP/CJM1
      B(IVEC) = 2.*P*U(I,NZ1) + (1.-2.*P)*U(I,NZ)
      IF (I.EQ.1) DX1 = XLOC(2) - XLOC(1)
      IF (I.NE.1) DX1 = XLOC(I) - XLOC(I-1)
      IF (I.NE.NX) DX2 = XLOC(I+1) - XLOC(I)
      IF (I.EQ.NX) DX2 = XLOC(NX) - XLOC(NX1)
      X1 = 2.*DX2/(DX1+DX2)
      X2 = 2. - X1
      R = PERMX(NZ)*DTIME/(WDENS*CJ*DX1*DX2)
      A(IVEC) = -R*X1
      B(IVEC) = 1. + 2.*R
      C(IVEC) = -R*X2
   42 CONTINUE
      IF (NBC2.EQ.1) C(1) = C(1) + A(1)
      IF (NBC3.EQ.1) A(IVEC) = A(IVEC) + C(IVEC)
      CALL TRIDG(A,B,C,D,X,IVEC,NOGO,ST1,ST2,ST3)
      IVEC = 0
      DO 43 I=ILIM1,ILIM2
      IVEC = IVEC + 1
   43 V(I,NZ) = X(IVEC)
   44 CONTINUE
C
C...NOW FOOTINGS; ASSUME SURFACE PERMEABLE
      IF (NBDRY(4).EQ.0) GO TO 48
      DZ = ZLOC(2) - ZLOC(1)
      CP = DTIME*PERMZ(2)/(WDENS*DZ*DZ)
      CT = 2.*(1.+PR(1))*(1.-2.*PR(1))
      DO 47 IFOOT=1,NFOOT
      ILIM1 = I1(IFOOT) + 1
      ILIM2 = I2(IFOOT) - 1
      IF (IFOOT.GT.1.AND.I2(IFOOT-1).EQ.I1(IFOOT)) ILIM1 = ILIM1 - 1
      IF (IFOOT.LT.NFOOT.AND.I1(IFOOT+1).EQ.I2(IFOOT))ILIM2=ILIM2+1
      IF (ILIM1.EQ.2.AND.NBC2.EQ.1) ILIM1 = 1
      IF (ILIM2.EQ.NX1.AND.NBC3.EQ.1) ILIM2 = NX
      IVEC =0
      DO 46 I=ILIM1,ILIM2
      IVEC = IVEC + 1
      CJ = CT/YM(I,1)
```

```
      CJP1 = CT/YH(I,2)
      Q = CQ/CJP1
      B(IVEC) = 2.*Q*U(I,2) + (1.-2.*Q)*U(I,1)
      IF (I.EQ.1) DX1 = XLOC(2) - XLOC(1)
      IF (I.NE.1) DX1 = XLOC(I) - XLOC(I-1)
      IF (I.EQ.NX) DX2 = XLOC(NX) - XLOC(NX1)
      IF (I.NE.NX) DX2 = XLOC(I+1) - XLOC(I)
      X1 = 2.*DX2/(DX1+DX2)
      X2 = 2. - X1
      R = PERMX(1)*DTIME/(WDENS*CJ*DX1*DX2)
      A(IVEC) = -R*X1
      B(IVEC) = 1. + 2.*R
      C(IVEC) = -R*X2
   40 CONTINUE
      IF (ILIM1.EQ.1) C(1) = C(1) + A(1)
      IF (ILIM2.EQ.NX) A(IVEC) = A(IVEC) + C(IVEC)
      CALL TRIDG(A,B,C,D,X,IVEC,NOGO,ST1,ST2,ST3)
      IVEC = 0
      DO 45 I=ILIM1,ILIM2
   45 V(I,1) = X(IVEC)
   47 CONTINUE
   48 CONTINUE
C
C.....SECOND EQUATION
      JLIM2 = NZ1 + NBC1
      NBC = NBDRY(5)
      IF (I1(1).EQ.1) NBC = NBDRY(4)
      IFOOT=1
      DO 25 I=2,NX1
C...CHECK IF FOOTING OR SURFACE
      IF (I.EQ.I1(IFOOT)+1) NBC = NBDRY(4)
      IF (I.NE.I2(IFOOT)) GO TO 18
      IF (I.EQ.NX) GO TO 18
      NBC = NBDRY(5)
      IF (IFOOT.LT.NFOOT) IFOOT = IFOOT + 1
   18 CONTINUE
      JLIM1 = 2 - NBC
      DX1 = XLOC(I) - XLOC(I-1)
      DX2 = XLOC(I+1) - XLOC(I)
      CR = DTIME/(WDENS*DX1*DX2)
      X1 = 2.*DX2/(DX1+DX2)
      X2=2.-X1
      JVEC = 0
      IF (NBC.EQ.0) GO TO 30
      JVEC = 1
      DZ = ZLOC(2) - ZLOC(1)
      CT = 2.*(1.+PR(1))*(1.-2.*PR(1))
      CJ = CT/YH(I,1)
      CJP1 = CT/YH(I,2)
      P = PERMZ(2)*DTIME/(WDENS*CJP1*DZ*DZ)
      R = PERMX(1)*CR/CJ
      B(1) = 1. + 2.*P
      C(1) = -2.*P
      D(1) = R*X1*V(I-1,1) + (1.-2.*R)*V(I,1) + R*X2*V(I+1,1)
   30 DO 15 J=2,NZ1
      JVEC = JVEC + 1
      Z1=2.*(ZLOC(J+1)-ZLOC(J))/(ZLOC(J+1)-ZLOC(J-1))
      Z2=2.-Z1
      CT = 2.*(1.+PR(J))*(1.-2.*PR(J))
      CJM1 = CT/YH(I,J-1)
```

```
      CJ    = CT/YM(I,J)
      CJP1 = CT/YM(I,J+1)
      T = CJM1*(ZLOC(J)-ZLOC(J-1)) + CJP1*(ZLOC(J+1)-ZLOC(J))
      PZ1 = PERMZ(J)
      PZ2 = PZ1
      IF (IFACE(J).EQ.0) GO TO 13
      PZ1 = PERMZ(J-1)
      PZ2 = PERMZ(J+1)
   13 CONTINUE
      P = 2.*PZ1*DTIME/(WDENS*(ZLOC(J)-ZLOC(J-1))*T)
      Q = 2.*PZ2*DTIME/(WDENS*(ZLOC(J+1)-ZLOC(J))*T)
      R = PERMX(J)*CR/CJ
      A(JVEC) = -P*Z1
      B(JVEC) = 1. + P + Q
      C(JVEC) = -Q*Z2
      D(JVEC) = R*X1*V(I-1,J) + (1.-2.*R)*V(I,J) + R*X2*V(I+1,J)
   15 CONTINUE
      IF (NBC1.EQ.0) GO TO 35
      JVEC = JVEC + 1
      DZ = ZLOC(NZ) - ZLOC(NZ1)
      CT = 2.*(1.+PR(NZ))*(1.-2.*PR(NZ))
      CJM1 = CT/YM(I,NZ1)
      CJ = CT/YM(I,NZ)
      Q = PERMZ(NZ1)*DTIME/(WDENS*CJM1*DZ*DZ)
      R = PERMX(NZ)*CR/CJ
      A(JVEC) = -2.*Q
      B(JVEC) = 1. + 2.*Q
      D(JVEC) = R*X1*V(I-1,NZ) + (1.-2.*R)*V(I,NZ) + R*X2*V(I+1,NZ)
   35 CONTINUE
      CALL TRIDG(A,B,C,D,X,JVEC,NOGO,ST1,ST2,ST3)
      JVEC = 0
      DO 21 J=JLIM1,JLIM2
      JVEC = JVEC + 1
   21 U(I,J) = X(JVEC)
   25 CONTINUE
C...SOLVE ALONG VERT. IMP. BDRIES
      IF (NBC2.EQ.0) GO TO 64
      NBC = NBDRY(5)
      IF (I1(1).EQ.1) NBC = NBDRY(4)
      JLIM1 = 2 - NBC
      DX = XLOC(2) - XLOC(1)
      CR = DTIME/(WDENS*DX*DX)
      JVEC = 0
      IF (NBC.EQ.0) GO TO 60
      JVEC = 1
      DZ = ZLOC(2) - ZLOC(1)
      CT = 2.*(1.+PR(1))*(1.-2.*PR(1))
      CJ = CT/YM(1,1)
      CJP1 = CT/YM(1,2)
      P = PERMZ(2)*DTIME/(WDENS*CJP1*DZ*DZ)
      R = PERMX(1)*CR/CJ
      B(1) = 1. + 2.*P
      C(1) = -2.*P
      D(1) = (1.-2.*R)*V(1,1) + 2.*R*V(2,1)
   60 DO 61 J=2,NZ1
      JVEC = JVEC + 1
      DZ1 = ZLOC(J) - ZLOC(J-1)
      DZ2 = ZLOC(J+1) - ZLOC(J)
      Z1 = 2.*DZ2/(DZ1+DZ2)
      Z2 = 2. - Z1
```

```
      CT = 2.*(1.+PR(J))*(1.-2.*PR(J))
      CJM1 = CT/YM(1,J-1)
      CJ = CT/YM(1,J)
      CJP1 = CT/YM(1,J+1)
      T = CJM1*DZ1 + CJP1*DZ2
      P = 2.*PERMZ(J-1)*DTIME/(WDENS*DZ1*T)
      Q = 2.*PERMZ(J+1)*DTIME/(WDENS*DZ2*T)
      R = PERMX(J)*CR/CJ
      A(JVEC) = -P*Z1
      B(JVEC) = 1. + P + Q
      C(JVEC) = -Q*Z2
      D(JVEC) = (1.-2.*R)*V(1,J) + 2.*R*V(2,J)
   61 CONTINUE
      IF (NBC1.EQ.0) GO TO 62
      JVEC = JVEC + 1
      DZ = ZLOC(NZ) - ZLOC(NZ1)
      CT = 2.*(1.+PR(NZ))*(1.-2.*PR(NZ))
      CJM1 = CT/YM(1,NZ1)
      CJ = CT/YM(1,NZ)
      Q = PERMZ(NZ1)*DTIME/(WDENS*CJM1*DZ*DZ)
      R = PERMX(NZ)*CR/CJ
      A(JVEC) = -2.*Q
      B(JVEC) = 1. + 2.*Q
      D(JVEC) = (1.-2.*R)*V(1,NZ) + 2.*R*V(2,NZ)
   62 CONTINUE
      CALL TRIDG(A,B,C,D,X,JVEC,NOGO,ST1,ST2,ST3)
      JVEC = 0
      DO 63 J=JLIM1,JLIM2
      JVEC = JVEC + 1
   63 U(1,J) = X(JVEC)
   64 CONTINUE
      IF (NBC3.EQ.0) GO TO 69
      NBC = NBDRY(5)
      IF (I2(NFOOT).EQ.NX) NBC = NBDRY(4)
      JLIM1 = 2 - NBC
      DX = XLOC(NX) - XLOC(NX1)
      CR = DTIME/(WDENS*DX*DX)
      JVEC = 0
      IF (NBC.EQ.0) GO TO 65
      JVEC = 1
      DZ = ZLOC(2) - ZLOC(1)
      CT = 2.*(1.+PR(1))*(1.-2.*PR(1))
      CJ = CT/YM(NX,1)
      CJP1 = CT/YM(NX,2)
      P = PERMZ(2)*DTIME/(WDENS*CJP1*DZ*DZ)
      R = PERMX(1)*CR/CJ
      B(1) = 1. + 2.*P
      C(1) = -2.*P
      D(1) = (1.-2.*R)*V(1,1) + 2.*R*V(2,1)
   65 DO 66 J=2,NZ1
      JVEC = JVEC + 1
      DZ1 = ZLOC(J) - ZLOC(J-1)
      DZ2 = ZLOC(J+1) - ZLOC(J)
      Z1 = 2.*DZ2/(DZ1+DZ2)
      Z2 = 2. - Z1
      CT = 2.*(1.+PR(J))*(1.-2.*PR(J))
      CJM1 = CT/YM(NX,J-1)
      CJ = CT/YM(NX,J)
      CJP1 = CT/YM(NX,J+1)
      T = CJM1*DZ1 + CJP1*DZ2
```

```
      P = 2.*PERMZ(J-1)*DTIME/(WDENS*DZ1*T)
      Q = 2.*PERMZ(J+1)*DTIME/(WDENS*DZ2*T)
      R = PERMX(J)*CR/CJ
      A(JVEC) = -P*Z1
      B(JVEC) = 1. + P + Q
      C(JVEC) = -Q*Z2
      D(JVEC) = (1.-2.*R)*V(NX,J) + 2.*R*V(NX1,J)
   66 CONTINUE
      IF (NBC1.EQ.0) GO TO 67
      JVEC = JVEC + 1
      DZ = ZLOC(NZ) - ZLOC(NZ1)
      CT = 2.*(1.+PR(NZ))*(1.-2.*PR(NZ))
      CJM1 = CT/YM(NX,NZ1)
      CJ = CT/YM(NX,NZ)
      Q = PERMZ(NZ1)*DTIME/(WDENS*CJM1*DZ*DZ)
      R = PERMX(NZ)*CR/CJ
      A(JVEC) = -2.*Q
      B(JVEC) = 1. + 2.*Q
      D(JVEC) = (1.-2.*R)*V(NX,NZ) + 2.*R*V(NX1,NZ)
   67 CONTINUE
      CALL TRIDG(A,B,C,D,X,JVEC,NDGO,ST1,ST2,ST3)
      JVEC = 0
      DO 68 J=JLIM1,JLIM2
      JVEC = JVEC + 1
   68 U(NX,J) = X(JVEC)
   69 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE FILON(DSZ,DSX,DTXZ,X1,X2,Q,NZ,NX,XLOC,ZLOC,
     1NX,NZ,ITYPE,EPS,I1,I2)
      IMPLICIT REAL*8(A-H,O-Z)
C...STRESS DISTRIBUTION FOR FINITE LAYER.
      DIMENSION DSZ(NZ,NX),DSX(NZ,NX),DTXZ(NZ,NX),XLOC(NX),ZLOC(NZ)
     1,F(8),G(8)
      DATA F/0.438,1.74,7.21,23.38,122.0,717.0,5050.0,40294.0/
      DATA G/0.9125,2.8042,5.75,24.825,119.,723.,5033.,40344./
      F0 = 0.527
C
      PI = 3.14159265D0
      QEPS = 1. - 1./((0.609522*EPS-0.0166826)*EPS+0.345663)
      D = ZLOC(NZ) - ZLOC(1)
      PID = PI*D
      XM = (X1 + X2)/2.
      B = XM - X1
      IF (ITYPE.NE.3) GO TO 3
      IF (X1.EQ.XLOC(1)) XM = X1
      IF (X2.EQ.XLOC(1)) XM = X2
      B = DMAX1(XM-X1,X2-XM)
    3 CONTINUE
C...MAKE DH SMALL COMPARED WITH MESH SIZE
      ID = I2 - I1 + 2
      DH = (X2-X1)/DFLOAT(5*ID)
      QH = Q*DH
      DO 12 JV=2,NZ
      ZZ = ZLOC(JV) - ZLOC(1)
      Z = 1. - ZZ/D
      DO 10 IV=1,NX
```

```
      SIGX = 0.
      SIGZ = 0.
      SIGT = 0.
      XLOCB = X1 + DH/2.
    5 CONTINUE
      XX = XLOCB - XLOC(IV)
      IF (DABS(XX).LT.1.0D-12) XX=0.0D0
      X = XX/D
      DB = (XH - XLOCB)/B
      QB = QH
      IF (ITYPE.EQ.1) QB=QB*(1.-DB)
      IF (ITYPE.EQ.2) QB=QB*(1.+DB)
      IF (ITYPE.EQ.3) QB=QB*QEPS/((1.-DABS(DB)**EPS)**(1./EPS))
      R = DSQRT(X*X + Z*Z)
      IF (R.LT.0.5D0) GO TO 6
      R4 = (XX*XX + ZZ*ZZ)**2
      ZR = ZZ/R4
      COEFF = 2.*QB/PI
      DSIGT = 0.0D0
      DSIGX = 0.D0
      DSIGZ = ZZ*ZZ*ZR
      IF (ZZ.GT.0.05D0*D) GO TO 8
      DSIGT = XX*ZZ*ZR
      DSIGX = XX*XX*ZR
      GO TO 8
    6 CONTINUE
      DSIGX = F0 - G(1)
      DSIGZ = F0 + G(1)
      DSIGT = 0.0D0
      PHI = PI/2.
      IF (Z.GT.1.0D-6) PHI = DATAN2(X,Z)
      NS = 1
      PP = 0.0D0
      COEFF = 1.0D0
      COUNT = 0.0D0
      DO 15 IS=1,7
      PP = PP + PHI
      CP = DCOS(PP)
      SP = DSIN(PP)
      COUNT = COUNT + 1.0D0
      NS = -NS
      COEFF = COEFF*R/COUNT
      IF (NS.EQ.1) GO TO 14
      ZF = COEFF*Z*F(IS+1)
      DSIGX = DSIGX + ZF*CP
      DSIGZ = DSIGZ - ZF*CP
      DSIGT = DSIGT - ZF*SP
      GO TO 15
   14 DSIGX = DSIGX + COEFF*CP*(F(IS)-G(IS+1))
      DSIGZ = DSIGZ + COEFF*CP*(F(IS)+G(IS+1))
      DSIGT = DSIGT - COEFF*SP*G(IS+1)
   15 CONTINUE
      COEFF = 2.*QB/PID
    8 SIGX = SIGX + COEFF*DSIGX
      SIGZ = SIGZ + COEFF*DSIGZ
      SIGT = SIGT + COEFF*DSIGT
      XLOCB = XLOCB + DH
      IF (XLOCB.LT.X2) GO TO 5
      DSX(JV,IV) = DSX(JV,IV) + SIGX
      DSZ(JV,IV) = DSZ(JV,IV) + SIGZ
```

```
      DTXZ(JV,IV) = DTXZ(JV,IV) + SIGT
   10 CONTINUE
   12 CONTINUE
C
C...STRESSES ON SURFACE
      DO 20 IV=1,NX
      X = XLOC(IV)
      IF (X.GT.X2) RETURN
      IF (X.LT.X1) GO TO 20
      QB = Q
      DB = (XH-X)/B
      COEFF = 1.0D0
      IF (IV.EQ.I1.OR.IV.EQ.I2) COEFF = 0.5
      IF (ITYPE.NE.3) GO TO 18
      IF (COEFF.EQ.1.0D0) GO TO 17
      IF (DB.GT.0.0D0) X = (X+XLOC(IV+1))/2.
      IF (DB.LT.0.0D0) X = (XLOC(IV-1)+X)/2.
      DB = (XH-X)/B
   17 QB = QB*QEPS/((1.-DABS(DB)**EPS)**(1./EPS))
   18 CONTINUE
      IF (ITYPE.EQ.1) QB=QB*(1.-DB)
      IF (ITYPE.EQ.2) QB=QB*(1.+DB)
      DSX(1,IV) = DSX(1,IV) + COEFF*QB
      DSZ(1,IV) = DSZ(1,IV) + COEFF*QB
   20 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE TRIDG(A,B,C,D,X,N,NOGO,ST1,ST2,ST3)
C - - REF: J.W.BLOSS;M.SC.DISS.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N),B(N),C(N),D(N),X(N),ST1(N),ST2(N),ST3(N)
      IF(NOGO.EQ.1) GO TO 20
      T2=C(1)/B(1)
      ST2(1)=T2
      DO 10 I=2,N
      T1=B(I)-A(I)*T2
      ST1(I)=T1
      T2=C(I)/T1
   10 ST2(I)=T2
   20 T3=D(1)/B(1)
      ST3(1)=T3
      DO 30 I=2,N
      T3=(D(I)-A(I)*T3)/ST1(I)
   30 ST3(I)=T3
      X(N)=ST3(N)
      K=N-1
      DO 40 J1=1,K
      I=N-J1
   40 X(I)=ST3(I)-ST2(I)*X(I+1)
      RETURN
      END
C
C
      SUBROUTINE PRINC(SX,SZ,TXZ,S1,S3)
C...FINDS PRINCIPAL STRESSES
      IMPLICIT REAL*8(A-H,O-Z)
C
      TEMP = DSQRT((SX-SZ)*(SX-SZ) + 4.D0*TXZ*TXZ)
```

```
      S1 = (SX + SZ + TEMP)/2.D0
      S3 = (SX + SZ - TEMP)/2.D0
      RETURN
      END
C
C
      SUBROUTINE OUTPUT(NX,NZ,SX,SZ,DISPZ,U,DISPF,MFOOT,NTIME,NITER,
     1MX,MZ,MFOOT,NSTEP,ROTN,MLDPT,INTER,DISPL,Q,ISUPN,TXZ,ICOLPT)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SX(MZ,MX),SZ(MZ,MX),DISPZ(MX,MZ),U(MX,MZ),DISPF(MFOOT)
     1,ROTN(MLDPT),DISPL(MLDPT),Q(MFOOT),TXZ(MZ,MX),ICOLPT(8)
C
      WRITE (6,600) NTIME,NITER,NSTEP
  600 FORMAT(1H0,/20X,80(1H*)//' OUTPUT FOR NTIME = ',I5,10X,
     1'NITER = ',I5,10X,'NSTEP = ',I5/1X,24(1H-)/)
C
C...OUTPUT FOOTING DISPLTS, AND STRUCTURE DISPLTS & ROTATIONS
      DO 5 I=1,NFOOT
      WRITE (6,605) I,DISPF(I),Q(I)
  605 FORMAT(1H ,'DISPLACEMENT AT FOOTING',I3,' = ',D12.4,
     1', PRESSURE = ',D12.4)
      IF (INTER.EQ.0) GO TO 5
      I1 = 2*I - 1
      WRITE (6,606) I1,DISPL(I1),ROTN(I1)
  606 FORMAT(1H ,64X,'DISPLACEMENT AT LOAD-PT',I3,' = ',
     1D11.4,', ROTATION = ',D11.4)
      IF (I.EQ.NFOOT) GO TO 5
      I1 = I1 + 1
      WRITE (6,606) I1,DISPL(I1),ROTN(I1)
    5 CONTINUE
C
      IF (ISUPN.LT.0.AND.ICOLPT(1).EQ.0) GO TO 12
      WRITE (6,610)
  610 FORMAT (1H0,'MESH-POINT: I    J ',5X,'STRESSES:      SX',13X,
     1'SZ',12X,'TXZ',10X,'DISPLACEMENTS:',5X,'PORE-PRESSURES:'/)
      NCOLPT = 1
      DO 15 I=1,NX
      IF (ISUPN.LT.0.AND.I.NE.ICOLPT(NCOLPT)) GO TO 15
      IF (NCOLPT.LT.7) NCOLPT = NCOLPT + 1
      DO 10 J=1,NZ
      WRITE (6,615) I,J,SX(J,I),SZ(J,I),TXZ(J,I),DISPZ(I,J),U(I,J)
  615 FORMAT(1H ,9X,2I5,14X,D12.4,2(3X,D12.4),6X,D12.4,8X,D12.4)
      IF (ISUPN.LT.-1) GO TO 15
   10 CONTINUE
      WRITE (6,625)
   15 CONTINUE
  625 FORMAT(1H )
   12 WRITE (6,620)
  620 FORMAT(1H ,20X,80(1H*))
      RETURN
      END
      SUBROUTINE INPUT(MX,MZ,MMAX,MFOOT,MLOAD,MLDPT,NX,NZ,NFOOT,NLOAD,
     1NDTMX,NDTIN,NTMAG,NTFAC,NTEND,LAYER,INTER,ISUPN,I1,IM,I2,XLOC,
     2ZLOC,YMINIT,PR,CC,PHI,PERMX,PERMZ,AP,BP,WDENS,GRAVY,WIDTH,DRMAX,
     3REACT,EPS,NBDRY,LDTIN,QEXT,LNTYP,EI,SXINIT,SZINIT,
     4ITYPE,ICOLPT,IFACE)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION I1(MFOOT),I2(MFOOT),XLOC(MX),ZLOC(MZ),YMINIT(MZ),
```

```
     1PR(MZ),CC(MZ),PHI(MZ),PERMX(MZ),PERMZ(MZ),NBDRY(5),LDTIM(MLOAD),
     2QEXT(MLDPT,MLOAD),LHTYP(MZ),EI(MFOOT),SXINIT(MZ),SZINIT(MZ),
     3CARD(8),KARD(8),IM(MFOOT),ITYPE(MFOOT),ICOLPT(8),IFACE(MZ)
C
      DO 4 I=1,MLDPT
      DO 4 J=1,MLOAD
    4 QEXT(I,J) = 0.0D0
C
      READ (5,501) NX,NZ,NFOOT,NLOAD,NDTMX,LAYER,INTER,ISTRE,ISUPN,
     1NDTIM,NTMAG,NTFAC,NTEND,ZSCAL
  501 FORMAT(12I5,I10,D10.3)
      IF (NFOOT.EQ.0) NFOOT = 1
      IF (NLOAD.EQ.0) NLOAD = 1
      IF (NDTMX.EQ.0) NDTMX = 99999
      IF (NTFAC.EQ.0) NTFAC = 10
      IF (NTMAG.EQ.0) NTMAG = 5
      IF (NDTIM.EQ.0) NDTIM = 2
      IF (ZSCAL.EQ.0.0D0) ZSCAL = 1.0D0
      WRITE (6,608) NX,NZ,NFOOT,NLOAD,NDTMX,NDTIM,NTMAG,NTFAC,NTEND,
     1ZSCAL
  608 FORMAT(1H0,'    NX    NZ NFOOT NLOAD NDTMX NDTIM NTMAG NTFAC'
     1,5X,'NTEND'/1X,8I6,I10//' ZSCAL = ',D10.3)
      WRITE (6,609) NDTIM,NTFAC,NTMAG,NDTMX,NTEND
  609 FORMAT(1H0,'TIMESTEPPING SCHEME: INITIAL TIMESTEP OF'
     1,I5,' INCREASES BY FACTOR OF',I5,' AFTER EVERY',I5,
     2' STEPS,'/1X,19(1H-),' UP TO A MAXIMUM OF',I6,
     3' STOPPING WHEN NTIME =',I10)
      IF (INTER.EQ.0) WRITE (6,623) INTER
      IF (INTER.NE.0) WRITE (6,624) INTER
      IF (ISUPN.LT.0) WRITE (6,626) ISUPN
      IF (ISTRE.NE.0) WRITE (6,627) ISTRE
      IF (LAYER.NE.0) WRITE(6,628) LAYER
  623 FORMAT(1H0,'INTER = ',I6,', SEPARATED FOOTINGS')
  624 FORMAT(1H0,'INTER = ',I6,', LINKED FOOTINGS')
  626 FORMAT(1H0,'ISUPN = ',I6,', ONLY STRUCTURAL VARIABLES OUTPUT')
  627 FORMAT(1H0,'ISTRE = ',I6,', NO INITIAL STRESSES')
  628 FORMAT(1H0,'LAYER = ',I6,'STRESS DISTRIBUTION FOR FINITE LAYER')
      IF (NX.GT.MX.OR.NZ.GT.MZ) GO TO 90
      MMAX = MAX0(MX,MZ)
      READ (5,516) (ICOLPT(I),I=1,8)
  516 FORMAT(8I5)
      IF (ISUPN.LT.0) WRITE (6,616) (ICOLPT(I),I=1,8)
  616 FORMAT(1H0,'OUTPUT FOR COLUMNS WITH I = ',8(I5,',')/1X,28(1H-)/)
      READ (5,502) (XLOC(I),I=1,NX)
  502 FORMAT(8D10.3)
      DO 5 I=1,NFOOT
    5 READ (5,503) I1(I),IM(I),I2(I),ITYPE(I)
  503 FORMAT(4I5)
      WRITE (6,602)
  602 FORMAT (1H0,2X,'I',4X,'XLOC(I)'/)
      IFOOT = 1
      DO 10 I=1,NX
      WRITE (6,603) I,XLOC(I)
  603 FORMAT (1H ,I3,1X,F10.3)
      IF (I.EQ.I1(IFOOT)) WRITE (6,604) IFOOT
  604 FORMAT (1H+,20X,'START OF FOOTING',I3)
      IF (I.EQ.IM(IFOOT)) WRITE (6,605) IFOOT,ITYPE(IFOOT)
  605 FORMAT(1H+,40X,'MID-POINT OF FOOTING',I3,', TYPE',I2)
      IF (I.NE.I2(IFOOT)) GO TO 10
      WRITE (6,606) IFOOT
```

```
  606 FORMAT (1H+,20X,'END OF FOOTING',I3)
      IF (IFOOT.NE.NFOOT) IFOOT=IFOOT+1
      IF (I.EQ.I1(IFOOT)) WRITE (6,619) IFOOT
  619 FORMAT(1H+,40X,'START OF FOOTING',I3)
   10 CONTINUE
C
      DO 11 J=1,NZ
      READ (5,504) ZLOC(J),PERMX(J),PERMZ(J),PR(J),IFACE(J)
      IF (J.EQ.1) GO TO 11
      IF (PERMX(J).EQ.0.0D0) PERMX(J) = PERMX(J-1)
      IF (PERMZ(J).EQ.0.0D0) PERMZ(J) = PERMZ(J-1)
      IF (PR(J).EQ.0.0D0) PR(J) = PR(J-1)
   11 ZLOC(J) = ZLOC(J)*ZSCAL
  504 FORMAT(4D10.3,I5)
      DO 12 J=1,NZ
      READ (5,514) LMTYP(J),YMINIT(J),CC(J),PHI(J)
      IF (J.EQ.1) GO TO 12
      IF (LMTYP(J).EQ.0) LMTYP(J) = LMTYP(J-1)
      IF (YMINIT(J).EQ.0.0D0) YMINIT(J) = YMINIT(J-1)
      IF (CC(J).EQ.0.0D0) CC(J) = CC(J-1)
      IF (PHI(J).EQ.0.0D0) PHI(J) = PHI(J-1)
   12 CONTINUE
  514 FORMAT(I5,5X,3D10.3)
      WRITE (6,607)
      DO 13 J=1,NZ
      IF (LMTYP(J).EQ.2) WRITE (6,618) J,IFACE(J),ZLOC(J),PERMX(J),
     1PERMZ(J),PR(J),LMTYP(J),YMINIT(J)
      IF (LMTYP(J).EQ.3)WRITE (6,618) J,IFACE(J),ZLOC(J),PERMX(J),
     1PERMZ(J),PR(J),LMTYP(J),YMINIT(J),CC(J),PHI(J)
      IF (LMTYP(J).EQ.4) WRITE (6,618) J,IFACE(J),ZLOC(J),PERMX(J),
     1PERMZ(J),PR(J),LMTYP(J),YMINIT(J),CC(J)
  618 FORMAT(1H ,I3,I2,2X,3(D10.4,5X),F5.3,I5,5X,2(D10.4,5X),F10.4)
   13 CONTINUE
  607 FORMAT(1H0,'LAYER DEPTHS & PROPERTIES'/1X,25(1H-)//' LAYER',4X,
     1'ZLOC(J)',7X,'PERMX(J)',7X,'PERMZ(J)',6X,'PR(J)',1X,'LMTYP',
     25X,'YMINIT(J)',8X,'CC(J)',10X,'PHI(J)'/)
C
      READ (5,532) AP,BP,WDENS,GRAVY,WIDTH,DRMAX,RFACT,EPS
  532 FORMAT(8D10.3)
      IF (BP.EQ.0.0D0) BP = 1.0D0
      IF (WDENS.EQ.0.0D0) WDENS = 9.81D0
      IF (GRAVY.EQ.0.0D0) GRAVY = 9.81D-3
      IF (WIDTH.EQ.0.0D0) WIDTH = 1.0D0
      IF (DRMAX.EQ.0.0D0) DRMAX = 1.0D-4
      IF (RFACT.EQ.0.0D0) RFACT = 0.8D0
      IF (EPS.EQ.0.0D0) EPS = 2.0D0
      WRITE (6,632) AP,BP,WDENS,GRAVY,WIDTH,DRMAX,RFACT,EPS
  632 FORMAT(1H0,'MISC. CONSTANTS'/1X,15(1H-)//
     1' PORE-PRESSURE CONSTANTS: AP = ',F6.3,', BP = ',F6.3/15X,
     2'WATER DENSITY = 'D10.3/15X,'GRAVITY FORCE = ',D10.3/
     315X,'FOOTING WIDTH = ',D10.3/23X,'DRMAX = ',D10.3/23X,
     4'RFACT = ',D10.3/25X,'EPS = ',D10.3/)
C
C...READ INITIAL EFFECTIVE STRESSES, IF NECESSARY
      IF (ISTRE.NE.0) GO TO 17
      WRITE (6,630)
  630 FORMAT(1H0,'INITIAL EFFECTIVE STRESSES'/1X,26(1H-)//
     15X,'J',11X,'SX',13X,'SZ'/)
      DO 16 J=1,NZ
      READ (5,531) SXJ,SZJ
```

```
  531 FORMAT(2D10.3)
      IF (J.EQ.1) GO TO 15
      IF (SXJ.EQ.0.0D0) SXJ = SXINIT(J-1)
      IF (SZJ.EQ.0.0D0) SZJ = SZINIT(J-1)
   15 CONTINUE
      WRITE (6,631) J,SXJ,SZJ
  631 FORMAT(1H ,I5,2(5X,D10.4))
      SXINIT(J) = SXJ
      SZINIT(J) = SZJ
   16 CONTINUE
   17 CONTINUE
C
C...AVERAGE THE SOIL PARAMETERS AT INTERNAL LAYER BOUNDARIES
      NZ1 = NZ -1
      DO 19 J=2,NZ1
      IF (IFACE(J).EQ.0) GO TO 19
      PERMX(J) = 0.5*(PERMX(J-1) + PERMZ(J+1))
      PERMZ(J) = 0.5*(PERMZ(J-1) + PERMZ(J+1))
      PR(J) = 0.5*(PR(J-1) + PR(J+1))
      YMINIT(J) = 0.5*(YMINIT(J-1) + YMINIT(J+1))
      CC(J) = 0.5*(CC(J-1) + CC(J+1))
      PHI(J) = 0.5*(PHI(J-1) + PHI(J+1))
      SXINIT(J) = 0.5*(SXINIT(J-1) + SXINIT(J+1))
      SZINIT(J) = 0.5*(SZINIT(J-1) + SZINIT(J+1))
   19 CONTINUE
C
      READ (5,508) (NBDRY(I),I=1,5)
  508 FORMAT(5I1)
      WRITE (6,612) (NBDRY(I),I=1,5)
  612 FORMAT(1H0,'BOUNDARY CONDITIONS: (0=PERM.,1=IMPERM.)'/1X,20(1H-)//
     1' BASE',3X,'LEFT SIDE',3X,'RIGHT SIDE',3X,'FOOTINGS',3X,
     2'SURFACE'/I4,I10,I13,I11,I10)
      WRITE (6,613)
  613 FORMAT(1H0,'LOAD SEQUENCE:'/1X,14(1H-)//' TIME',5X,'LOAD-PT',5X,
     1'MAGNITUDE'/)
      DO 20 L=1,NLOAD
   24 READ (5,510) (CARD(I),I=1,8)
  510 FORMAT(3D10.3)
      READ (5,509) LCARD,LDTIM(L),(KARD(I),I=1,8)
  509 FORMAT(I1,I4,8I5)
      DO 25 I=1,8
      IF (KARD(I).EQ.0) GO TO 26
      IKARD = KARD(I)
      QEXT(IKARD,L) = CARD(I)
   25 WRITE (6,614) LDTIM(L),IKARD,CARD(I)
  614 FORMAT(1H ,I4,7X,I3,6X,D10.3)
   26 IF (LCARD.EQ.0) GO TO 24
      WRITE (6,615)
  615 FORMAT(1H )
   20 CONTINUE
      IF (INTER.EQ.0) RETURN
      NBAY = NFOOT - 1
      READ (5,510) (EI(I),I=1,NBAY)
      WRITE (6,611) (I,EI(I),I=1,NBAY)
  611 FORMAT(1H0,'EI FOR BAYS:'/1X,12(1H-)//
     1I20(' EI(',I2,') = ',D10.3/))
      RETURN
   90 WRITE (6,690)
  690 FORMAT(1H0,'*** - FATAL ERROR: MAX. DIMENSIONS EXCEEDED')
      STOP
```

```
      END
C
C
      SUBROUTINE YMHYP(MZ,J,YM,RFACT,PHI,CC,S1,S3,DS)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PHI(MZ),CC(MZ)
C
C...FINDS CURRENT YOUNG'S MOD FOR HYPERBOLIC-LAW MESH-PT
C
      IF (DS.LE.0.0D0) GO TO 5
      PHIRD = PHI(J)/57.2958D0
      SINP = DSIN(PHIRD)
      COSP = DCOS(PHIRD)
      FACT = 1. - RFACT*(S1-S3)*(1.-SINP)/(2.*S3*SINP+2.*CC(J)*COSP)
      IF (FACT.LT.0.1D0) FACT=0.1D0
      YM = YM*FACT*FACT
    5 RETURN
      END
C
C
      SUBROUTINE YMLIN(MZ,J,YM,CC,DS)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CC(MZ)
C
C...FINDS CURRENT YOUNG'S MODULUS FOR 'BILINEAR-LAW' MESH-PT
C
      IF (DS.GT.0.0D0) GO TO 5
      YM = YM*CC(J)
    5 RETURN
      END
C
C
      SUBROUTINE STRUCT(QEXT,Q,DISPF,ILOAD,NFOOT,NLDPT,NFOOT,
     1NLOAD,GRHS,NSIZE,ROTN,DISPL,VLHS,VRHS)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION QEXT(NLDPT,NLOAD),Q(NFOOT),
     1DISPF(NFOOT),GRHS(NSIZE,NSIZE),VLHS(NSIZE),VRHS(NSIZE),
     2ROTN(NLDPT),DISPL(NLDPT)
C
C...FIRST PUT KNOWN QUATITIES INTO VRHS.
      NSIZE = 2*(2*NFOOT-1)
      II = 1
      DO 5 I=1,NFOOT
      VRHS(II) = DISPF(I)
      VRHS(II+1) = 0.0D0
      IF (I.EQ.NFOOT) GO TO 5
      VRHS(II+2) = QEXT(2*I,ILOAD)
      VRHS(II+3) = 0.0D0
      II = II + 4
    5 CONTINUE
C...NOW PREMULTIPLY VRHS BY GRHS, PUT RESULT IN VLHS
      DO 20 I=1,NSIZE
      SUM = 0.0D0
      DO 15 J=1,NSIZE
   15 SUM = SUM + GRHS(I,J)*VRHS(J)
      VLHS(I) = SUM
   20 CONTINUE
      II = 1
```

```
      DO 25 I=1,NFOOT
      Q(I) = VLHS(II)
      DISPL(2*I-1) = DISPF(I)
      ROTN(2*I-1) = VLHS(II+1)
      IF (I.EQ.NFOOT) GO TO 25
      DISPL(2*I) = VLHS(II+2)
      ROTN(2*I) = VLHS(II+3)
      II = II + 4
   25 CONTINUE
C...ADD EXT. LLOADS ON FOOTINGS TO - REACTIONS TO GET Q
      DO 35 I=1,NFOOT
   35 Q(I) = QEXT(2*I-1,ILOAD) - Q(I)
      RETURN
      END
C
C
      SUBROUTINE STIFF(MX,NFOOT,MSIZE,NFOOT,EI,IN,XLOC,GRHS,GSTIF)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION EI(NFOOT),IN(NFOOT),XLOC(MX),GSTIF(MSIZE,MSIZE),
     1GRHS(MSIZE,MSIZE)
C
C.....SETS UP BEAM STIFFNESS & RHS MATRICES, THEN FORMS (A**-1)*B.
      MSIZE=2*(2*NFOOT-1)
      IF (MSIZE.GT.MSIZE) GO TO 50
      DO 5 I=1,MSIZE
      DO 5 J=1,MSIZE
      GSTIF(I,J)=0.D0
      GRHS(I,J)=0.D0
    5 CONTINUE
C
C.....FIRST BAY.
      IFOOT=1
      GSTIF(1,1)=1.0
      GRHS(2,2)=-1.
      GRHS(3,3)=-1.
      GRHS(4,4)=-1.
      IN1 = IN(1)
      IN2 = IN(2)
      XL = (XLOC(IN2) - XLOC(IN1))/2.
      C=-2*EI(1)/XL
      GSTIF(2,2)=2.*C
      GSTIF(2,4) = C
      GSTIF(4,2) = C
      GSTIF(4,4) = 4.*C
      GSTIF(4,6) = C
      C = -3.*C/XL
      GSTIF(1,2) = -C
      GSTIF(1,3) = 2.*C/XL
      GSTIF(1,4) = -C
      GSTIF(2,3) = C
      GSTIF(3,2) = C
      GSTIF(3,3) = -4.*C/XL
      GSTIF(3,6) = -C
      GRHS(2,1) = C
      GRHS(3,1) = -2.*C/XL
      GRHS(3,5) = -2.*C/XL
      GRHS(4,1) = C
      GRHS(4,5) = -C
      GRHS(1,1) = 2.*C/XL
```

```
      JJ = 2
      II = 4
C
C.....I'TH BAY.
   10 IFOOT = IFOOT + 1
      INFT1 = IN(IFOOT-1)
      INFT = IN(IFOOT)
      INFT2 = IN(IFOOT+1)
      IF (IFOOT.EQ.NFOOT) GO TO 15
      GSTIF(II+1,JJ+3) = 1.
      GRHS(II+2,JJ+4) = -1.
      GRHS(II+3,JJ+5) = -1.
      GRHS(II+4,JJ+6) = -1.
      X = XLOC(INFT)
      XL1 = (X - XLOC(INFT1))/2.
      XL = (XLOC(INFT2) - X)/2.
      C1 = -2.*EI(IFOOT-1)/XL1
      C = -2.*EI(IFOOT)/XL
      GSTIF(II+2,JJ+2) = C1
      GSTIF(II+2,JJ+4) = 2.*(C1+C)
      GSTIF(II+2,JJ+6) = C
      GSTIF(II+4,JJ+4) = C
      GSTIF(II+4,JJ+6) = 4.*C
      GSTIF(II+4,JJ+8) = C
      C1 = -3.*C1/XL1
      C = -3.*C/XL
      GSTIF(II+1,JJ+1) = 2.*C1/XL1
      GSTIF(II+1,JJ+2) = C1
      GSTIF(II+1,JJ+4) = C1-C
      GSTIF(II+1,JJ+5) = 2.*C/XL
      GSTIF(II+1,JJ+6) = -C
      GSTIF(II+2,JJ+1) = -C1
      GSTIF(II+2,JJ+5) = C
      GSTIF(II+3,JJ+4) = C
      GSTIF(II+3,JJ+5) = -4.*C/XL
      GSTIF(II+3,JJ+8) = -C
      GRHS(II+1,JJ+3) = 2.*(C1/XL1 + C/XL)
      GRHS(II+2,JJ+3) = -C1 + C
      GRHS(II+3,JJ+3) = -2.*C/XL
      GRHS(II+3,JJ+7) = -2.*C/XL
      GRHS(II+4,JJ+3) = C
      GRHS(II+4,JJ+7) = -C
C
      II = II + 4
      JJ = JJ + 4
      GO TO 10
C
C.....FINAL FOOTING.
   15 GSTIF(II+1,JJ+3) = 1.
      GSTIF(II+2,JJ+4) = -1.
      XL1 = (XLOC(INFT) - XLOC(INFT1))/2.
      C1 = -2.*EI(IFOOT-1)/XL1
      GSTIF(II+2,JJ+2) = C1
      GSTIF(II+2,JJ+4) = 2.*C1
      C1 = -3.*C1/XL1
      GSTIF(II+1,JJ+1) = 2.*C1/XL1
      GSTIF(II+1,JJ+2) = C1
      GSTIF(II+1,JJ+4) = C1
      GSTIF(II+2,JJ+1) = -C1
      GRHS(II+1,JJ+3) = 2.*C1/XL1
```

```
      GRHS(II+2,JJ+3) = -C1
      II = II + 2
C
C
C...NOW REDUCE GSTIF TO IDENTITY MATRIX,
C...DOING SAME OPS TO GRHS, WHICH BCOMES REQD MATRIX
      DO 40 I=1,NSIZE
      P = GSTIF(I,I)
      IF (DABS(P).LT.1.0D-6) WRITE (6,598) I,P
      DO 35 J=1,NSIZE
      GSTIF(I,J) = GSTIF(I,J)/P
   35 GRHS(I,J) = GRHS(I,J)/P
      DO 40 K=1,NSIZE
      IF (K.EQ.I) GO TO 40
      Q = GSTIF(K,I)
      IF (Q.EQ.0.0D0) GO TO 40
      DO 45 L=1,NSIZE
      GSTIF(K,L) = GSTIF(K,L) - Q*GSTIF(I,L)
   45 GRHS(K,L) = GRHS(K,L) -Q*GRHS(I,L)
   40 CONTINUE
  598 FORMAT (1H ,'WARNING: PIVOT IN ROW',I5,' OF GSTIF=',D10.3)
      RETURN
   50 WRITE (6,599) NSIZE,NSIZE
      STOP
  599 FORMAT(1H ,'STOPPED IN STIFF: NSIZE=',I5,', EXCEEDING MSIZE=',I5)
      END
```

1322 LINES

APPENDIX B - Program Specification: FINETIM

The program FINETIM is essentially an adaptation of the finite element package FINEPAK to model time-dependent consolidation. The intending user is therefore referred in the first instance to the User Instructions for FINEPAK, printed by the Centre for Numerical Methods in Engineering of the University College of Swansea. We here indicate the differences between the two programs, as they affect the user. The theory used in FINETIM is given in Chapter 4 of this thesis.

Main Program

The main or master program, named FINTIM(F), is a little more complex than in FINEPAK; its basic structure is shown in the flow diagram in Fig.B1.

There is one extra dimension which must be included in the DATA statement at the start, viz. MPFIX. This is used in subroutine SMOOTH, and it must exceed MFIXV plus the number of pore-pressure d.o.f.s in the model.

There is one new main subroutine (SMOOTH), and extra satellite subroutines YMHYP and YMLIN to evaluate the Young's modulus of a non-linear elastic material. The FINEPAK subroutine MOD has been renamed MOD1 to avoid confusion with the Fortran standard function MOD, which may cause errors in some machines.

Subroutine INPUT

The input formats are now given, with asterisks marking the differences from FINEPAK. Real variables are in the D10.3 format, which is more flexible than FINEPAK's F10.3.

Type	No.	Format	Variables read
C1	1	11I5,I10	NPOIN,NELEM,NFIXV,MSTYP,NDIME, NDOFM,NGAUS,NSTRE,NDTIM*,NTMAG*, NTEND*.
C2	1	11I5,I10	NPROS,NPROP,NSSET,NDOFD*,NTFAC*, LDTYP,NINCS,NITER,NBSET,NPSET, NLOAD*, ISMOTH*.
C3	1	8D 10.3	PIVAL,PWBM*,RFACT*,ZSCAL*,SCONV, PSCAL*,DTONE*,THETA*.

CONT'D .....

..... CONT'D

Type	No.	Format	Variables read
G1	as reqd	I1,I4,13I5	LCARD,NSIDE,LNUM(elt.no.),(LNODS(LNUM,N),N=1,NNODE).Note 6-noded quadrilaterals must have mid-side nodes listed in positions 2 & 5.
G2	as reqd	I1,I4,I5, 3D10.3	LCARD,NDOFN,N(node no.),(COORD(N,M),M=1,3)(COORD(N,3)defaults to 1.0 for 2D problems)
G3	as reqd	I1,I9,14I5	LCARD,KODE,list of nodes to which KODE applies(1=fixed,0=free d.o.f; for pore pressure d.o.f.s*, 1=permeable,Ø=impermeable boundary)
LM	as reqd	I1,I4,15I5	LCARD,LMTD,LMT,up to 14 elts to which LMT applies. LMT=1 structural element*(no pore-pressures). LMT=2 linear elastic soil elt LMT=3 hyperbolic law soil elt* LMT=4 'bilinear' law soil elt*
M1	as reqd	I1,I4,3X, 6I2,6D10.3	LCARD,NSET, up to 6 property component nos., up to 6 property component values. Property components are: 1. Young's modulus E 2. Poisson's ratio ν 3. Cohesion*C' for LMT=3, E_{unload}/E_{load} for LMT = 4 4. Angle of friction*Ø' for LMT=3 5. Porosity*n 6.) Relative permeabilities* 7.) For 2D problems use only 6 & 7 8.) for k_x/γ_w and k_z/γ_w respectively.
M2	as reqd	I1,I4,I2,I3, 14I5	LCARD,NSETD,LN,NSET,up to 14 elements (LN=1) or nodes (LN=2) associated with NSET
I1*	as reqd (1)	1I1,D9.3, 5D10.3	LCARD,(STRES(I),I=1,NSTRE).Initial effective stresses. Omit if NSSET= 0
	as reqd (1)	1 16I5	elements for which these initial stresses apply. Omit if NSSET=0
O1	as reqd	I1,I9,14I5	LCARD,LORQL,list of up to 14 elts to which LORQL applies.

CONT'D .....

..... CONT'D

Type	No.	Format	Variables read
O2*	1	16I5	(INDPT(I),I=1,16)List of up to 16 nodes at which load/displt output is required. If output at all nodes required, leave card blank.
LD1*	NLOAD	I1,I4,2I5	LCARD,LDTIM(I),NBSTV(I),NPSTV(I) Time of application,NBSET and NPSET for I'th load,I=1,NLOAD. If NLOAD =1,omit these cards and use NBSET & NPSET on C1 card,LDTIM assumed=0.

We now list the variables used in cards C1,C2,C3. Note that some variables from FINEPAK have new defaults. The number of degrees of freedom for nodes not listed in G2 cards defaults to a new variable NDOFD, rather than to NDOFM as in FINEPAK. This enables a problem to be solved using composite elements, then again using quadratic-elements-with-smoothing with only minor changes to the data cards.

NCONS	Name	Description	Default	Max
1	NPOIN	No of nodes in mesh	none	MPOIN
2	NELEM	No of elements in mesh	none	MELEM
3	NFIXV	No of fixed d.o.f.s.	none	MFIXV
4	MSTYP	Mesh type (0=plane strain; 1=plane stress;2=axi-symm.; 3=3D)	0	3
5	NDIME	No of dimensions	2	3
6	NODEM	No of nodes in largest element (ie max.NNODE)	8	MNODE
7	NDOFM	Max. no. of d.o.f. per node	3*	6
8	NGAUS	Order of Gaussian integration rule	2	2
9	NSTRE	No of stress components (not input; assigned values 3,3,4, 6 as MSTYPE=0,1,2,3 resp.)		6
10	NDTIM*	Initial timestep	2	
11	NTMAG*	No of steps after which NDTIM is increased	5	

CONT'D .....

..... CONT'D

NCONS	Name	Description	Default	Max
12	NTEND*	Time at which run ends (note input as I10)	0	
13	NPROS	No of material property sets	none	MSETS
14	NPROP	Highest mat.prop. component no in a set	8*	MCOMP
15	NSSET	No of sets of initial stress data (only necessary to specify 0 or > 0	0	
16	NDOFD*	Default no. for d.o.f.s if not given in G2 cards	3	6
17	NTFAC*	Factor by which NDTIM is increased every NTMAG Steps	10	
18	LDTYP	Not used in FINETIM	0	2
19	NINCS	No. of load increments (used to apply initial load only)	1	100
20	NITER	Not used in FINETIM	1	100
21	NBSET	No. of B cards in load 1 (not used if NLOAD > 1)	0	
22	NPSET	No. of P cards in load 1 (not used if NLOAD > 1)	0	
23	NLOAD*	No. of loads to be applied	1	
24	ISMOTH*	> 0 for smoothing of initial pore pressures	0	
CMISC				
1	PIVAL	Min. allowable pivot in FRONT	10^{-6}	
2	PWBM*	Pore water bulk modulus	1.8×10^{6}	
3	RFACT*	Adjustment factor R_f for hyperbolic law elements	0.8	
4	ZSCAL*	Scaling factor by which all z-coords in G2 cards are multplied (used in studying one problem for different thichnesses of soil layer)	1.0	
5	SCONV	Sign convention:-1= compression +ve, 1=compression -ve.	-1.0	
6	PSCAL*	Scaling factor applied to porepressure to improve scaling of stiffness matrix(effect seen in max. and min. pivot sizes used in FRONT and output)	1.0	

CONT'D .....

..... CONT'D

NCONS	Name	Description	Default	Max.
7	DTONE*	Small timestep used for initial undrained solution	0.0	
8	THETA*	Value of θ used in time-discretization	0.0	

Timestepping Scheme

The timestepping scheme is the same as that employed in FDTIM. The initial timestep NDTIM is increased by a factor of NTFAC after every NTMAG steps; when further loading occurs the timestep reverts to the initial NDTIM. For the initial solution a very small timestep should be used; this is specified as DTONE in the C3 cards. The default of zero for the time discretization variable THETA is not recommended; the usual value to use is 0.5.

Pore-pressures

The pore-pressure at a node is taken as the final degree of freedom. If the node lies on a permeable boundary, this d.o.f. should be fixed; if on an impervious boundary it is left free and the natural boundary conditions apply. A node on a soil-structure interface may have a pore-pressure d.o.f.; this will be ignored in compiling the stiffness matrix for the structural element.

Loading

The input of loads is unchanged from FINEPAK. (See FINEPAK manual under subroutine LOAD). When more than one loading is required, the load data cards are simply placed in order at the end of the input data; the numbers of B and P cards for each load are specified in the arrays NBSTV and NPSTV. Note that after the first load, later loads must specify the total forces acting including previous loads, not just the new load being applied (as in FDTIM). On the other hand, specified displacements given in the initial load do not need to be repeated.

Subroutine SMOOTH

Variables in	Variables out
NELEM, NPOIN, NSDIS,LSIDN, ASDIS, COORD,NPNDF, SPDIS, SHAPE, ENCOD, LNODS,LNODN,LMTYP,NFVAR	NBDIS, NBVAR, BSDIS.

Smooths the initial pore-pressures for two-dimensional soil elements, returning them as specified displacements in BSDIS (which also includes the displacements specified in ASDIS) for resolution by FRONT.

Implementation

The program, like FINEPAK, uses three workfiles. These use devices 17,18,and 19, as required by the System 4 DATAD module WORK 1. The main subroutines (including SMOOTH) are kept in the file SUBTIM(F), and the satellite subroutines which are common to FINETIM and FINEPAK are in the file SUBCOM(F). A MULTIJOB task to run the program on any SWUCN machine except the SWURCC one, must therefore include the command:

```
**INCLUDE NETLIB:DATAD.WORK1
```

For running on SWURCC, the workfiles are set up by the macros

```
USE(UNIT=17)
USE(UNIT=18)
USE(UNIT=19)
```

The SWURCC workfiles are limited to 2280 kilobytes, equivalent to 285,000 double precision numbers. This puts a limit on the size of problem which can be treated viz.

MELEM x MEVAB x MFRON < 285,000

restricting MELEM to about 200. Running of the program in single precision arithmetic has been found to cause errors in the last two significant figures of output displacements.

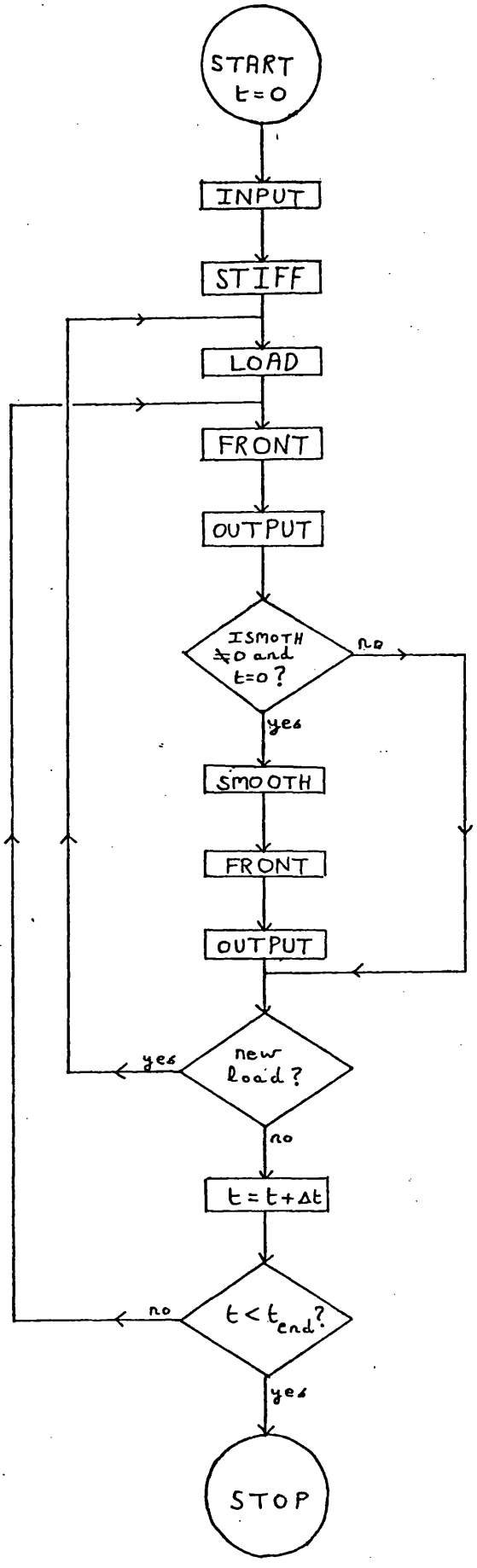


Fig. B1 Simplified flow diagram: FINETIM master program

Program Listing

Program FINTIM(F)

This is the master segment of FINETIM

```
C      MASTER FINETIM
C...TIME-DEPENDENT VERSION OF FINLPAK - BATH, 1ST FEB 1980
C...REQUIRES MAIN S/R'S IN SUBTIM(F) & SATELLITE ONES IN SUBCOM(F)
C...DIMENSIONED FOR 200 ELEMENTS, 600 POINTS
C
       IMPLICIT REAL*8 (A-H,O-Z)
       DIMENSION NCONS(24),LNODS(200,8),LNODN(200),LSIDN(200),NPNDF(650)
      1,NFVAR(250),LMTYP(200),LPROS(200,8),LOREQ(200)
      2,MACVA(75),LOCEL(24),NDEST(24)
      3,COORD(650,3),ENCOD(3,8),ASLOD(1950),ASDIS(1950),SPDIS(250)
      4,PROPS(9,10),ENPRO(10,8),GPROP(10),SHAPE(8),DERIV(3,8)
      5,CARTD(3,8),BMATX(6,24),DBMAT(6,24),ENDIS(24)
      6,ESTIF(24,24),GSTIF(2900),VECRV(75),GLOAD(75),EQUAT(75)
      7,STRSL(200,6),LBOPS(200,8),BFOPS(9,10),CMISC(8),TITLE(16)
       DIMENSION ASLD1(1950),ASLD2(1950),LDTIM(10),LPORN(8),PERM(3)
      1,UMATX(3,24),MMATX(8,8),SMATX(8,8),XMATX(24,24),YMATX(24,24)
      2,NBSTV(10),NPSTV(10),ENCDP(3,8),INDPT(16)
      3,SUMS(650,2),ASRHS(600),BSDIS(600),NDVAR(600),ASDS1(1950)
      4,PSTSL(200,4)
C
       EQUIVALENCE(NCONS(1),NPOIN),(NCONS(2),NELEM),(NCONS(3),NFIXV)
      1,(NCONS(4),NSTYP),(NCONS(5),NDIME),(NCONS(6),NODEM)
      2,(NCONS(7),NDOFM),(NCONS(8),NGAUS),(NCONS(9),NSTRE)
      3,(NCONS(10),NDTIM),(NCONS(11),NTMAG),(NCONS(12),NTEND)
      3,(NCONS(13),NPROS),(NCONS(14),NPROP)
      4,(NCONS(15),NSSET),(NCONS(16),NDOFD),(NCONS(17),NTFAC)
      5,(NCONS(18),LDTYP),(NCONS(19),NINCS),(NCONS(20),NITER)
      6,(NCONS(21),NBSET),(NCONS(22),NPSET)
      7,(NCONS(23),NLOAD),(NCONS(24),ISMOTH)
       EQUIVALENCE (CMISC(1),PIVAL),(CMISC(2),PWBM ),(CMISC(3),RFACT)
      1,(CMISC(4),ZSCAL),(CMISC(5),SCONV),(CMISC(6),PSCAL)
      2,(CMISC(7),DTONE),(CMISC(8),THETA)
C...THESE EQUIVALENCE STATEMENTS INCLUDE ALL THE CONTROL PARAMETERS
C...READ IN SUBROUTINE INPUT, NOT ALL ARE NEEDED FOR LINEAR APPLICATIONS
C
```

```
      DATA MTOTV,MPOIN,MELEM,MFIXV,MEVAB,MNODE,MSETS,MCOMP,MFRON,MSTIF
     1    /1950, 650 , 200 , 250 ,  24 ,   8 ,   9 ,  10 ,  75 , 2900/
C...DATA STATEMENT MUST TALLY WITH DIMENSION STATEMENT. CONFUSION IS
C...AVOIDED IF DIFFERENT VALUES USED FOR MTOTV,MPOIN, ETC.
      DATA MPFIX
     1    / 600 /
C
      WRITE(6,600)
    5 READ (5,501) LCARD,TITLE
      WRITE(6,601) TITLE
      IF (LCARD.EQ.0) GO TO 5
      WRITE(6,602)
      ILOAD = 0
      MTIME = 0
      MSTEP = 0
      DO 15 I=1,MELEM
      DO 10 J=1,4
   10 PSTSL(I,J) = 0.0D0
      DO 15 J=1,6
      STRSL(I,J) = 0.0D0
   15 CONTINUE
      DO 20 I=1,MTOTV
      ASLD1(I) = 0.0D0
      ASLD2(I) = 0.0D0
      ASDIS(I) = 0.0D0
      ASDS1(I) = 0.0D0
   20 CONTINUE
C
      CALL INPUT(MPOIN,MELEM,MFIXV,MNODE,MSETS,MCOMP,
     1MCONS,LNODS,LNODN,LSIDN,NPNDF,NFVAR,LMTYP,LPROS,STRSL,
     2COORD,PROPS,LOREQ,CMISC,
     3LDTIN,NBSTV,NPSTV,PSTSL,INDPT)
C...ISUPN=1 TO STOP CALCS OF STRSL & PSTSL IN UNSMOOTHED PORE-PRESS CASE
      ISUPN = 1
      IF (ISMOTH.EQ.0) ISUPN = 0
      MTINIT = MDTIM
C
      THT = THETA/(1.D0 - THETA)
      IRSOL = 0
      CALL STIFF(MPOIN,MELEM,MEVAB,MNODE,MSETS,MCOMP,
     1MELEM,NDIME,NGAUS,NSTRE,NPROP,MSTYP,
     2LNODS,LNODN,LSIDN,NPNDF,LMTYP,LPROS,COORD,ENCOD,SCONV,
     3PROPS,ENPRO,GPROP,SHAPE,DERIV,CARTD,BMATX,DBMAT,ESTIF
     4,LPORM,PERM,NTIME,UMATX,HMATX,SMATX,XMATX
     5,YMATX,PSCAL,PWBM,ENCDP)
C
C...READ DATA FOR NEXT LOAD
   25 CONTINUE
      IF (NLOAD.EQ.1) GO TO 30
      NBSET = NBSTV(ILOAD+1)
      NPSET = NPSTV(ILOAD+1)
   30 CONTINUE
C
      CALL LOAD (MTOTV,MPOIN,MELEM,MFIXV,MEVAB,MNODE,MSETS,MCOMP,
     1MPOIN,MELEM,MFIXV,NSDIS,NDIME,NGAUS,MSTYP,NODEM,NBSET,NPSET,
     2LNODS,LNODN,LSIDN,NPNDF,NFVAR,LBOPS,BFOPS,SCONV,
     3COORD,ENCOD,ENPRO,SHAPE,DERIV,CARTD,ASLD2,SPDIS,NTIME,PSCAL)
C...NEW NODAL FORCES NOW IN ASLD2
C
      ILOAD = ILOAD + 1
```

```
      DTIME = DTONE
      INC = 1
      NSTEP = 0
      NDTIM = NTINIT
C
   35 DINCS = DFLOAT(INC)/DFLOAT(NINCS)
C...FORM R.H.S. VECTOR FOR STIFFNESS EQN
      DO 40 I=1,NTOTV
      ASLD1(I) = TMT*ASLD1(I) + ASLD2(I)*DINCS
   40 CONTINUE
C
      CALL FRONT(NTOTV,NPOIN,NELEM,NFIXV,NEVAB,NNODE,NFRON,NSTIF,
     1NPOIN,NELEM,NSDIS,LNODS,LNODN,NPNDF,NFVAR,ASLOD,ASDIS,SPDIS,
     2VECRV,ESTIF,GSTIF,GLOAD,EQUAT,NACVA,LOCEL,NDEST,PIVAL,IRSOL
     3,ASLD1,XMATX,YMATX,THETA,NTIME,LNTYP,NSTYP,NGAUS,SHAPE,DERIV
     4,DTIME,LPORN,PSTSL,ASRHS,LPROS,PROPS,MSETS,MCOMP,RFACT,LSIDN)
C...FRONT RETURNS R.H.S. VECTOR PLUS REACTIONS IN ASLD1
C
      WRITE (6,603) NTIME,IRSOL,NSTEP
      CALL OUTPUT(NTOTV,NPOIN,NELEM,NEVAB,NNODE,NSETS,MCOMP,
     1NPOIN,NELEM,NDIME,NGAUS,NSTRE,NSTYP,NDOFN,NPROP,
     2LNODS,LNODN,LSIDN,NPNDF,LNTYP,LOREQ,LPROS,COORD,ENCOD,SHAPE,
     3PROPS,ENPRO,DERIV,CARTD,BMATX,ASLD1,ASDIS,ENDIS,ASDS1,
     4SCONV,0.D0,0.D0,INC,0,ISUPN,PSCAL,PSTSL,STRSL,RFACT,INDPT)
      ISUPN = 0
C
C...APPLY SMOOTHING TO INITIAL PORE-PRESSURES, IF REQD
      IF (NSTEP.GT.0) GO TO 43
      IF (ISMOTH.EQ.0) GO TO 43
      CALL SMOOTH(NTOTV,NPOIN,NFIXV,NSTYP,NELEM,NNODE,NELEM,
     1NPOIN,NSDIS,LSIDN,ASDIS,BSDIS,COORD,NPNDF,SPDIS,SUMS,SHAPE,ENCOD,
     2LNODS,LNODN,LNTYP,NFVAR,NBVAR,NBDIS,NPFIX)
      CALL FRONT(NTOTV,NPOIN,NELEM,NPFIX,NEVAB,NNODE,NFRON,NSTIF,
     1NPOIN,NELEM,NBDIS,LNODS,LNODN,NPNDF,NBVAR,ASLOD,ASDIS,BSDIS,
     2VECRV,ESTIF,GSTIF,GLOAD,EQUAT,NACVA,LOCEL,NDEST,PIVAL,IRSOL,
     3ASLOD,XMATX,YMATX,THETA,NTIME,LNTYP,NSTYP,NGAUS,SHAPE,DERIV,
     4DTONE,LPORN,PSTSL,ASRHS,LPROS,PROPS,MSETS,MCOMP,RFACT,LSIDN)
      WRITE (6,604)
      CALL OUTPUT(NTOTV,NPOIN,NELEM,NEVAB,NNODE,NSETS,MCOMP,
     1NPOIN,NELEM,NDIME,NGAUS,NSTRE,NSTYP,NDOFN,NPROP,
     2LNODS,LNODN,LSIDN,NPNDF,LNTYP,LOREQ,LPROS,COORD,ENCOD,SHAPE,
     3PROPS,ENPRO,DERIV,CARTD,BMATX,ASLOD,ASDIS,ENDIS,ASDS1,
     4SCONV,0.D0,0.D0,1.0,0,PSCAL,PSTSL,STRSL,RFACT,INDPT)
   43 CONTINUE
C
C...PLACE FORCE & DISPLT VECTORS FOR THIS TIMESTEP IN ASLD1,ASDS1
      DO 45 I=1,NTOTV
      ASLD1(I) = ASLD2(I)*DINCS
      ASDS1(I) = ASDIS(I)
   45 CONTINUE
C
C...ADD NEXT INITIAL LOAD INCREMENT, IF NECC
      IF (NTIME.GT.0) GO TO 50
      INC = INC + 1
      IF (INC.LE.NINCS) GO TO 35
      INC = 1
      NINCS = 1
   50 CONTINUE
C
C...INCREASE TIMESTEP IF REQUIRED
```

```
      IF (MOD(NSTEP,NTMAG).EQ.0.AND.NSTEP.GT.0) NDTIM=NDTIM*NTFAC
      NSTEP = NSTEP + 1
C...CHECK IF TIME FOR NEW LOAD
      IF (NTIME.GT.0.AND.NTIME.EQ.LDTIM(ILOAD+1)) GO TO 25
      NTIME = NTIME + NDTIM
      DTIME = DFLOAT(NDTIM)
      IF (NTIME.LE.NTEND) GO TO 35
C
      STOP
  501 FORMAT (I1,16A4)
  600 FORMAT(1H0,//16H PROGRAM FINETIM,//1X,85(1H*)/1X,1H*,83X,1H*)
  601 FORMAT (1X,1H*,2X,16A4,17X,1H*)
  602 FORMAT(1X,1H*,83X,1H*,/1X,85(1H*))
  603 FORMAT(1H ,/8H TIME = ,I5,5X,9H IRSOL = ,I1,5X,
     1 9H NSTEP = ,I5/1X,12(1H-))
  604 FORMAT(1H0,25X,10(1H*),25H SMOOTHED PORE-PRESSURES ,
     1 10(1H*)/)
      END
C
C
```

176 LINES

Program Listing

Program SUBTIM(F)

This contains the main subroutines for FINETIM, viz: INPUT, STIFF, LOAD, FRONT, OUTPUT & SMOOTH. A vertical line on the left-hand side indicates the principal additions & changes from FINEPAK.

```
C...SUBTIM(F).....MAIN S/R'S FOR FINTIM(F)
C...USES SATELLITE S/R'S IN SUBCOM(F)
C...ADAPTED FROM FINEPAK MARK 3 BY M. B. REED
C...IMPLEMENTED ON SWUCH MULTIJOB....1ST FEB 1980
C
C              * * * * *
C
      SUBROUTINE INPUT(MPOIN,MELEM,MFIXV,MNODE,MSETS,MCOMP,
     1MCONS,LNODS,LNODN,LSIDN,NPNDF,NFVAR,LMTYP,LPROS,STRSL,
     2COORD,PROPS,LOREQ,CMISC,
     3LDTIM,NBSTV,NPSTV,PSTSL,INDPT)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MCONS(24),LNODS(MELEM,MNODE),LNODN(MELEM),LSIDN(MELEM)
     1,NPNDF(MPOIN),COORD(MPOIN,3),STRSL(MELEM,6),STRES(6),PSTRS(12)
     2,NFVAR(MFIXV),LMTYP(MELEM),LPROS(MELEM,MNODE),PROPS(MSETS,MCOMP)
     3,LOREQ(MELEM),CMISC(8)
     4,MCONS(24),KARD(16),CARD(20)
     5,MCONS(32),HEAD1(3),HEAD2(3),FIX(6)
     6,LDTIM(10),NBSTV(10),NPSTV(10),PSTSL(MELEM,4),INDPT(16)
C
      DATA BLANK,FIXED,PI2/5H     ,5HFIXED,6.283185307/
      DATA DEGRAD/0.0174532925/
      DATA IBRAC,JBRAC/1H(,1H)/
      DATA HEAD1/5H (1) ,5H (2) ,5H (3) /
      DATA HEAD2/5H N/A ,5HTHICK,5H     /
      DATA MCONS/0,0,0,3,  3,0,  6,  8,6,99999,99999,99999,
     1          0,0,0,6,100,2,100,100,0,    0,   50,99999/
      DATA MCONS/
     16HNPOIN ,6HNELEM ,6HNFIXV ,6HNSTYP ,6HNDIME ,6HNODEM ,
     26HNDOFN ,6HNGAUS ,6HNSTRE ,6HNDTIM ,6HNTMAG ,6HNTEND ,
     36HNPROS ,6HNPROP ,6HNSSET ,6HNDOFD ,6HNTFAC ,6HLDTYP ,
     46HNINCS ,6HNITER ,6HNBSET ,6HNPSET ,6HNLOAD ,6HISMOTH,
     56HPIVAL ,6HPWDH  ,6HRFACT ,6HZSCAL ,6HSCONV ,6HPSCAL ,
     66HDTONE ,6HTHETA /
```

```
C...NCONS PROVIDES ALTERNATIVE TO FORMATS
C
C...THIS SUBROUTINE READS AND CHECKS ALL INPUT FOR P.STRAIN, P.STRESS,
C...AXI-SYMMETRIC, 3D, OR BEAM PROBLEMS. ELEMENTS MAY BE LINE(FRAMEWORKS
C...AND BENDING), TRIANGULAR, OR 4 SIDED. THEY MAY BE LINEAR OR PARABOL-
C...IC. THE FIRST 6 NUMBERS IN ARGUMENT LIST, IE TO NCOMP, MUST BE ASS-
C...IGNED VALUES TO FIX MAX ARRAY SIZES. IF FIXED DIMENSIONS ARE ALTERED
C...THE DATA NCONS STATEMENT MUST BE ALTERED TOO.
C
      NCONS(1) = MPOIN
      NCONS(2) = MELEM
      NCONS(3) = MFIXV
      NCONS(6) = MNODE
      NCONS(13)= MSETS
      NCONS(14)= MCOMP
      NCONS(15)= MSETS
      NCONS(16)= MCOMP
      NCONS(21) = MSETS
      NCONS(22) = MSETS
C
      ISTOP = 0
C
C
C...CONTROL PARAMETERS (2 CARDS - C1,C2)
      READ(5,401) (NCONS(I),I=1,24)
C...IF NDIME=3, MSTYP MUST ALSO BE 3
      IF(NCONS(5).EQ.3) NCONS(4) = 3
C...ASSIGN VALUES 3,4 OR 6 TO MSTRE DEPENDING ON MSTYP
      MSTYP = NCONS(4)
      NCONS(9) = 3
      IF(MSTYP.GE.2) NCONS(9) = 2*MSTYP
C...ASSIGN DEFAULTS
      IF(NCONS( 5).EQ.0) NCONS( 5) = 2
      IF(NCONS( 6).EQ.0) NCONS( 6) = 8
      IF (NCONS( 7).EQ.0) NCONS( 7) = 3
      IF(NCONS( 8).EQ.0) NCONS( 8) = 2
      IF (NCONS(10).EQ.0) NCONS(10) = 2
      IF (NCONS(11).EQ.0) NCONS(11) = 5
      IF(NCONS(13).EQ.0) NCONS(13) = 1
      IF (NCONS(14).EQ.0) NCONS(14) = 8
      IF (NCONS(16).EQ.0) NCONS(16) = 3
      IF (NCONS(17).EQ.0) NCONS(17) = 10
      IF(NCONS(19).EQ.0) NCONS(19) = 1
      IF(NCONS(20).EQ.0) NCONS(20) = 1
      IF (NCONS(23).EQ.0) NCONS(23) = 1
      WRITE(6,601)(NCONS(I),NCONS(I),NCONS(I+12),NCONS(I+12),I=1,12)
C...CHECK FOR ERRORS
      DO 10 I=1,24
      IF (I.EQ.12.OR.I.EQ.24) GO TO 10
      IF(NCONS(I).GE.0.AND.NCONS(I).LE.NCONS(I)) GO TO 10
      ISTOP = ISTOP + 1
      WRITE(6,601) NCONS(I)
   10 CONTINUE
      MPOIN = NCONS( 1)
      MELEM = NCONS( 2)
      MFIXV = NCONS( 3)
      MSTYP = NCONS( 4)
      NDIME = NCONS( 5)
      NODEM = NCONS( 6)
      NDOFM = NCONS( 7)
```

```
      NGAUS = NCONS( 8)
      NSTRE = NCONS( 9)
C     NDTIM = NCONS(10)       NOT USED IN THIS SUBROUTINE
C     NTMAG = NCONS(11)     NOT USED IN THIS SUBROUTINE
C     NTEND = NCONS(12)       NOT USED IN THIS SUBROUTINE
      NPROS = NCONS(13)
      NPROP = NCONS(14)
      NSSET = NCONS(15)
      NDOFD = NCONS(16)
C     NTFAC = NCONS(17)       NOT USED IN THIS SUBROUTINE
C     LDTYP = NCONS(18)       NOT USED IN THIS SUBROUTINE
      NINCS = NCONS(19)
      NBSET = NCONS(21)
      NPSET = NCONS(22)
      NLOAD = NCONS(23)
      NITER = NCONS(20)
      READ(5,411) (CMISC(I),I=1,8)
      IF (CMISC(1).EQ.0.0D0) CMISC(1) = 1.0D-6
      IF (CMISC(2).EQ.0.0D0) CMISC(2) = 0.18D+7
      IF (CMISC(3).EQ.0.0D0) CMISC(3) = 0.8D0
      IF (CMISC(4).EQ.0.0D0) CMISC(4) = 1.0D0
      IF (CMISC(5).EQ.0.0D0) CMISC(5) = -1.0D0
      IF (CMISC(6).EQ.0.0D0) CMISC(6) = 1.0D0
      WRITE(6,611)(WCONS(I+24),CMISC(I),I=1,8)
      ZSCAL = CMISC(4)
C
C
C...ELEMENT CONNECTIONS (NELEM CARDS - G1)
      WRITE(6,620)(IBRAC,N,JBRAC,N=1,NODEM)
      NM12 = NODEM
      IF(NODEM.GT.12) NM12=12
      K=0
   15 K=K+1
      READ(5,421)LCARD,NSIDE,LNUM,(LNODS(LNUM,N),N=1,NM12)
      IF(NODEM.GT.12)READ(5,422)(LNODS(LNUM,N),N=13,NODEM)
      IF(NSIDE.EQ.0) NSIDE = 4
      IF(NSIDE.GE.1.AND.NSIDE.LE.6) GO TO 20
      ISTOP = ISTOP+1
      WRITE(6,820) NSIDE
   20 LSIDN(LNUM) = NSIDE
C...FIND NNODE AND STORE IN LNODN()
      NNODE = 0
      DO 21 N=1,NODEM
      IF(LNODS(LNUM,N).EQ.0) GO TO 21
      NNODE = NNODE + 1
   21 CONTINUE
      LNODN(LNUM) = NNODE
      WRITE(6,621)LNUM,NSIDE,(LNODS(LNUM,N),N=1,NNODE)
      IF(LNUM.GT.0.AND.LNUM.LE.NELEM) GO TO 22
      ISTOP = ISTOP+1
      WRITE(6,821)
   22 DO 23 N=1,NNODE
      IF(LNODS(LNUM,N).GT.0.AND.LNODS(LNUM,N).LE.NPOIN) GO TO 23
      ISTOP = ISTOP+1
      WRITE(6,822)N,LNUM
   23 CONTINUE
C...CHECK FOR REPETITION OF NODE IN ELEMENT
      DO 25 N=2,NNODE
      NP = LNODS(LNUM,N)
      N1 = N-1
```

```
      DO 24 NN=1,N1
      IF(LNODS(LNUM,NN).NE.NP) GO TO 24
      ISTOP = ISTOP+1
      WRITE(6,823) NP,LNUM
   24 CONTINUE
   25 CONTINUE
      IF(LCARD.EQ.0) GO TO 15
      IF(K.EQ.NELEM) GO TO 26
C...WRONG NUMBER OF G1 CARDS
      ISTOP = ISTOP+1
      WRITE(6,824)K,NELEM
C...CHECK IF ANY NODES OMITTED
   26 DO 28 NP=1,NPOIN
      DO 27 L=1,NELEM
      NNODE = LNODN(L)
      DO 27 N=1,NNODE
      IF(LNODS(L,N).EQ.NP) GO TO 28
   27 CONTINUE
      WRITE(6,825) NP
   28 CONTINUE
C
C...NODAL COORDS AND D.OF F. (NB MIDSIDE NODES MAY BE OMITTED AND D.OF F
C...DEFAULTS TO NDOFD) - (G2)
      DO 35 N=1,NPOIN
      NPNDF(N) = NDOFD
      DO 35 M=1,3
   35 COORD(N,M) = 0.D0
      WRITE(6,840) (HEAD1(I),I=1,NDIME),(HEAD2(I),I=NDIME,3)
   40 READ(5,441)LCARD,NDOFN,N,(COORD(N,M),M=1,3)
      IF(NDOFN.GT.0) NPNDF(N) = NDOFN
      IF(N.GT.0.AND.N.LE.NPOIN) GO TO 45
      ISTOP = ISTOP+1
      WRITE(6,841)N
   45 IF(LCARD.EQ.0) GO TO 40
C...INTERPOLATE MIDSIDE NODE VALUES IF NEC.
C...WE USE CARD() FOR TEMP STORAGE OF CO-ORDS
      DO 50 L=1,NELEM
      NNODE = LNODN(L)
      NSIDE = LSIDN(L)
      DO 49 M=1,3
      DO 47 N=1,NNODE
      NP = LNODS(L,N)
   47 CARD(N) = COORD(NP,M)
      CALL MSHVAL(NNODE,NSIDE,CARD)
      DO 48 N=1,NNODE
      NP = LNODS(L,N)
   48 COORD(NP,M) = CARD(N)
   49 CONTINUE
   50 CONTINUE
C...MULT. Z-COORDS BY ZSCAL IF ZSCAL.NE.1
      IF (ZSCAL.EQ.1.0D0) GO TO 54
      NZ = 2
      IF (MSTYP.EQ.3) NZ = 3
      DO 55 N=1,NPOIN
      COORD(N,NZ) = COORD(N,NZ) * ZSCAL
   55 CONTINUE
   54 CONTINUE
C...SOME MORE MESSING AROUND WITH COORD TO EASE LIFE LATER
      IF(NDIME.EQ.3) GO TO 59
      IF(MSTYP.EQ.2) GO TO 57
```

```
C...P. STRESS OR STRAIN, PUT DEFAULT UNIT THICKNESS IN COORD(N,3)
      DO 56 N=1,NPOIN
      IF(COORD(N,3).EQ.0.D0) COORD(N,3) = 1.D0
   56 CONTINUE
      GO TO 59
C...AXI-SYM, PUT 2*PI*R IN COORD(N,3)
   57 DO 58 N=1,NPOIN
      R = COORD(N,1)
      IF(R.GE.0.D0) GO TO 58
      ISTOP = ISTOP+1
      WRITE(6,842) N
   58 COORD(N,3) = PI2*R
C...PRINT COORDINATES AND NODAL D.OF F.
   59 DO 60 N=1,NPOIN
   60 WRITE(6,641) N,NPNDF(N),(COORD(N,M),M=1,3)
C
C...BOUNDARY CONSTRAINTS (G3)
      WRITE(6,660) (IBRAC,N,JBRAC,N=1,NDOFN)
      NFV=0
C...READ CONSTRAINT CODE FOLLOWED BY LIST OF NODES TO WHICH IT APPLIES
   62 READ(5,461)LCARD,KODE,(KARD(I),I=1,14)
      DO 70 I=1,14
      N = KARD(I)
      IF(N.EQ.0) GO TO 70
      IF(N.GT.0.AND.N.LE.NPOIN) GO TO 65
      ISTOP = ISTOP+1
      WRITE(6,841)N
C...SORT OUT KODE AND STORE FIXED VARIABLE NO. IN NFVAR()
   65 KOD = KODE
      NDOFN = NPNDF(N)
      DO 67 M=1,NDOFN
      KD = KOD/10
      K = KOD - KD*10
      KOD = KD
      IF(K.EQ.0) GO TO 67
      NFV=NFV+1
      M1 = NDOFN + 1 - M
      NFVAR(NFV) = NPOSN(N,M1,NPNDF)
C...NPOSN IS A FUNCTION SUBROUTINE
   67 CONTINUE
   70 CONTINUE
      IF(LCARD.EQ.C) GO TO 62
C...CHECK NO. OF G3 CARDS CORRECT
      IF(NFV.EQ.NFIXV) GO TO 72
      WRITE(6,862) NFV
      NFIXV = NFV
      NCONS(3) = NFV
      IF(NFV.LE.NFIXV) GO TO 72
      ISTOP = ISTOP+1
      WRITE(6,801) NCONS(3)
      GO TO 76
C...PRINT BOUNDARY CONSTRAINT DATA
   72 NN=0
      DO 75 N=1,NPOIN
      NON = 0
      NDOFN = NPNDF(N)
      DO 74 M=1,NDOFN
      FIX(M) = BLANK
      NN=NN+1
      DO 73 J=1,NFIXV
```

```
      IF(NN.NE.NFVAR(J)) GO TO 73
      FIX(N) = FIXED
      NON = 1
   73 CONTINUE
   74 CONTINUE
      IF(NON.EQ.1)
     1WRITE(6,661) N,(FIX(N),N=1,NDOFN)
   75 CONTINUE
   76 CONTINUE
C
C...ELEMENT MATERIAL TYPE (LIN. ELASTIC, VISCO-PLASTIC, ETC.) (LM)
      WRITE(6,680)
      DO 79 L=1,NELEM
   79 LMTYP(L) = 0
C...READ TYPE (LMT) FOLLOWED BY LIST OF ELEMENTS TO WHICH IT APPLIES.
C...LMTD IS DEFAULT KEPT FROM LAST CARD OF SET.
   80 READ(5,481)LCARD,LMTD,LMT,(KARD(I),I=1,14)
      DO 85 I=1,14
      L = KARD(I)
      IF(L.EQ.0) GO TO 85
      IF(L.GT.0.AND.L.LE.NELEM) GO TO 83
      ISTOP = ISTOP+1
      WRITE(6,881)L
   83 LMTYP(L) = LMT
   85 CONTINUE
      IF(LCARD.EQ.0) GO TO 80
C...ASSIGN DEFAULTS
      IF (LMTD.EQ.0) LMTD=2
      DO 90 L=1,NELEM
      IF(LMTYP(L).GT.0) GO TO 90
      LMTYP(L) = LMTD
      IF(LMTD.GT.0) GO TO 90
      ISTOP = ISTOP+1
      WRITE(6,882)L
   90 CONTINUE
C...PRINT DISTRIBUTION OF ELEMENT TYPES
      DO 95 L=1,NELEM
      WRITE(6,681) L,LMTYP(L)
   95 CONTINUE
C
C...INPUT MAT. PROPERTIES (M1,M2)
      CALL SETVAL(NELEM,MNODE,MSETS,MCOMP,MPOIN,NELEM,NODEM,ISTOP,
     1MPROS,MPROP,LNODS,LNODM,LPROS,PROPS,0,
     25HMAT. ,5HPROPE,5HRTIES,5H (M1),5H (M2))
C
      IF (NSSET.EQ.0) GO TO 160
      NDSTR = 2
      IF (NSTYP.GE.2) NDSTR = 3
C...PUT INITIAL EFFECTIVE STRESSES INTO STRSL
C...ALSO CALC. PRINC. EFFECTIVE STRESSES & PUT INTO PSTSL.
      WRITE (6,670) (I,I=1,NSTRE)
   96 READ (5,470) LCARD,(STRES(I),I=1,NSTRE)
      WRITE (6,671) (STRES(I),I=1,NSTRE)
      CALL PRINC(1,NSTRE,STRES,PSTRS)
      READ (5,471) (KARD(I),I=1,16)
      DO 97 I=1,16
      IMAX = I
      IKARD = KARD(I)
      IF (IKARD.EQ.0) GO TO 98
      IF (IKARD.LT.0.OR.IKARD.GT.NELEM) WRITE (6,881) IKARD
```

```
      DO 94 J=1,NSTRE
   94 STRSL(IKARD,J) = STRES(J)
      MINOR = 2
      IF (NSTRE.EQ.6) MINOR=3
      PSTSL(IKARD,3) = PSTRS(1)
      PSTSL(IKARD,4) = PSTRS(MINOR)
   97 CONTINUE
   98 IMAX = IMAX - 1
      WRITE (6,672) (KARD(I),I=1,IMAX)
      IF (LCARD.EQ.0) GO TO 96
C
C
C...ELEMENT OUTPUT REQUIREMENTS (01)
  160 WRITE (6,770)
      DO 169 L=1,NELEM
  169 LOREQ(L) = 0
C...LORQL IS OUTPUT CONTROL CODE (SEE OUTPUT). ELEMENTS TO WHICH IT
C...APPLIES READ FROM KARD(). DEFAULT IS NO OUTPUT.
  170 READ(5,461)LCARD,LORQL,(KARD(I),I=1,14)
      DO 175 I=1,14
      L = KARD(I)
      IF(L.EQ.0) GO TO 175
      IF(L.GT.0.AND.L.LE.NELEM)GO TO 173
      ISTOP = ISTOP+1
      WRITE(6,881) L
      GO TO 175
  173 LOREQ(L) = LORQL
  175 CONTINUE
      IF(LCARD.EQ.0) GO TO 170
C...PRINT ELEMENT OUTPUT REQUIREMENTS
      DO 177 L=1,NELEM
  177 WRITE(6,681) L,LOREQ(L)
      WRITE(6,772)
C...READ INDPT - NODES WHERE OUTPUT REQUIRED.
      READ (5,465) LCARD,(INDPT(I),I=1,16)
      IF (INDPT(1).EQ.0) GO TO 189
      WRITE (6,665)
      DO 185 I=1,16
      IF (INDPT(I).NE.0) WRITE(6,666) INDPT(I)
  185 CONTINUE
  189 CONTINUE
C
C
C...READ LOAD TIMES & NBSETS & NPSETS IF MORE THAN ONE LOAD.
      LDTIM(1) = 0
      NBSTV(1) = NBSET
      NPSTV(1) = NPSET
      IF (NLOAD.EQ.1) GO TO 205
      WRITE (6,793)
      I = 1
  201 READ (5,493) LCARD,LDTIM(I),NBSTV(I),NPSTV(I)
      WRITE (6,794) LDTIM(I),NBSTV(I),NPSTV(I)
      I = I + 1
      IF (LCARD.EQ.0) GO TO 201
C
C
  205 IF (ISTOP.EQ.0) RETURN
      WRITE(6,1000)ISTOP
      STOP
C
```

```
C...INPUT FORMATS
  401 FORMAT(11I5,I10)
  411 FORMAT(8D10.3)
  421 FORMAT(I1,I4,13I5)
  422 FORMAT(10X,8I5)
  441 FORMAT(I1,I4,I5,3D10.3)
  461 FORMAT(I1,I9,14I5)
  465 FORMAT(I1,I4,15I5)
  470 FORMAT(I1,D9.3,5D10.3)
  471 FORMAT(16I5)
  481 FORMAT(I1,I4,15I5)
  495 FORMAT(I1,I4,2I5)
C
C...OUTPUT FORMATS
  601 FORMAT(//19H CONTROL PARAMETERS,/1X,18(1H-),
     1/ 3X,4H(C1),18X,4H(C2),11(/4X,2(A6,1H=,I5,10X)),/4X,
     22(A6,1H=,I10,5X))
  611 FORMAT(//29H MISC CONTROL PARAMETERS (C3),/1X,28(1H-),
     1 3(/4X,A6,1H=,D13.5))
  620 FORMAT(//25H ELEMENT CONNECTIONS (G1),/1X,24(1H-),
     1/ 2X,7HELEMENT, 8H  NO. OF,11X,20HNODES AROUND ELEMENT,
     2/ 4X,3HNO.,4X,5HSIDES,4X,20(A1,I2,A1,1X))
  621 FORMAT(I6,I8,4X,20I5)
  640 FORMAT(//24H NODAL CO-ORDINATES (G2),/1X,23(1H-),
     1/ 3X,4HNODE,3X,5HNDOFN, 6(8X,A5))
  641 FORMAT(1X,I5,I7,5X,3D13.5)
  660 FORMAT(//23H NODAL CONSTRAINTS (G3),/1X,22(1H-);
     1/ 3X, 4HNODE,6X,8HD. OF F.,/ 4X,3HNO.,5X,6(A1,I1,A1,4X))
  661 FORMAT(1X,I5,5X,6(A5,2X))
  665 FORMAT(//31H OUTPUT ONLY AT FOLLOWING NODES/1X,30(1H-))
  666 FORMAT(1H ,I5)
  670 FORMAT(//27H INITIAL EFFECTIVE STRESSES/1X,26(1H-)
     1//9X,8HELEMENTS,14X,6(6X,6HSTRESS,I2))
  671 FORMAT(1H0,34X,D10.3,5(5X,D10.3))
  672 FORMAT(1H+,5(1X,5I5/))
  680 FORMAT(//27H ELEMENT MATERIAL TYPE (LM),/1X,26(1H-),
     1/ 3X,16HELEMENT  LMTYP())
  681 FORMAT(2I8)
  687 FORMAT(1H+,32X,15I5)
  770 FORMAT(//33H ELEMENT OUTPUT REQUIREMENTS (O1),/1X,32(1H-),
     1/ 3X,16HELEMENT  LOREQ())
  772 FORMAT( 3X,63HLOREQ(L) IS OUTPUT CONTROL CODE. IF ZERO NO OUTPUT I
     1N ELEMENT L)
  780 FORMAT(//35H INCREMENT OUTPUT REQUIREMENTS (O2),/1X,34(1H-),
     1/40H OUTPUT ORDERED FOR FOLLOWING INCREMENTS)
  781 FORMAT(20X,I5)
  790 FORMAT(//35H ITERATION OUTPUT REQUIREMENTS (O3),/1X,34(1H-),
     1/40H OUTPUT ORDERED FOR FOLLOWING ITERATIONS)
  793 FORMAT(//34H LOADS APPLIED AT FOLLOWING TIMES:,5X,5HNBSET,
     35X,5HNPSET/1X,33(1H-),5X,5(1H-),5X,5(1H-))
  794 FORMAT(1H ,21X,I5,10X,I5,5X,I5)
C
C...ERROR MESSAGE FORMATS
  801 FORMAT(18H *** FATAL ERROR, ,A6,24HOUTSIDE PERMITTED LIMITS)
  820 FORMAT(37H *** FATAL ERROR, CANNOT HAVE NSIDE =,I3)
  821 FORMAT(49H *** FATAL ERROR, ELEMENT NO. NEGATIVE OR TOO BIG)
  822 FORMAT(38H *** FATAL ERROR, NODE NO. IN POSITION,I3,
     111H IN ELEMENT,I4,23H IS NEGATIVE OR TOO BIG)
  823 FORMAT(22H *** FATAL ERROR, NODE,I4,20H REPEATED IN ELEMENT,I4)
  824 FORMAT(18H *** FATAL ERROR, ,I4,25H G1 CARDS READ. SHOULD BE,I4)
```

```
  825 FORMAT(22H *** WARNING, NODE NO.,I4,20H OMITTED IN G1 CARDS)
  841 FORMAT(18H *** FATAL ERROR, ,I5,28H IS NOT A PERMITTED NODE NO.)
  842 FORMAT(54H *** FATAL ERROR, AXI-SYM MESH HAS NEG. RADIUS AT NODE,
     1 I4)
  862 FORMAT(39H *** WARNING - NO. OF FIXED VALUES READ OFF G3 CARDS DIF
     1FERS FROM NFIXV. NFIXV ALTERED TO,I4)
  881 FORMAT(18H *** FATAL ERROR, ,I5,31H IS NOT A PERMITTED ELEMENT NO.
     1)
  882 FORMAT(63H *** FATAL ERROR, NO MATERIAL TYPE HAS BEEN ASSIGNED TO
     1ELEMENT,I4)
  981 FORMAT(18H *** FATAL ERROR, ,I5,29H IS NOT A PERMITTED INCREMENT)
  991 FORMAT(18H *** FATAL ERROR, ,I5,29H IS NOT A PERMITTED ITERATION)
 1000 FORMAT(///I5,46H FATAL ERRORS IN INPUT. BETTER LUCK NEXT TIME.)
      END
C
C
C
      SUBROUTINE STIFF(MPOIN,MELEM,MEVAB,MNODE,MSETS,MCOMP,
     1MELEM,NDIME,NGAUS,NSTRE,NPROP,MSTYP,
     2LNODS,LNODN,LSIDN,MPNDF,LMTYP,LPROS,COORD,ENCOD,SCONV,
     3PROPS,ENPRO,GPROP,SHAPE,DERIV,CARTD,BMATX,DBMAT,ESTIF
     4,LPORN,PERM,MTIME,UMATX,MMATX,SMATX,XMATX,YMATX
     5,PSCAL,PWBM,ENCDP)
C...ENCOD TO ESTIF INCL. INTERNAL TO STIFF AND ITS SATELLITES
C...EVALUATES ELEMENT STIFFNESSES AND WRITES THEM ON FILE 17
C...ALSO PUTS OTHER ELEMENT DATA ON FILE 3
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION LNODS(MELEM,MNODE),LNODN(MELEM),LSIDN(MELEM)
     1,MPNDF(MPOIN),LMTYP(MELEM),LPROS(MELEM,MNODE),LTYPS(9)
     2,COORD(MPOIN,3),ENCOD(3,MNODE),GPCOD(3),GPLOC(3,3),GPWTS(8)
     3,PROPS(MSETS,MCOMP),ENPRO(MCOMP,MNODE),GPROP(MCOMP)
     4,SHAPE(MNODE),DERIV(3,MNODE),CARTD(3,MNODE)
     5,BMATX(6,MEVAB),DMATX(6,6),DBMAT(6,MEVAB)
     6,ESTIF(MEVAB,MEVAB),VALUE(20)
     7,LPORN(MNODE),PERM(3),UMATX(MNODE,MEVAB)
     8,MMATX(MNODE,MNODE),SMATX(MNODE,MNODE),XMATX(MEVAB,MEVAB)
     9,YMATX(MEVAB,MEVAB),ENCDP(3,MNODE)
C
      DATA SMALL/1.0E-12/
      DATA LTYPS/21,31,33,44,63,64,84,86,206/
C...LTYPS ENCODES 9 ELEMENT TYPES.(FIRST FIG. IS NO.OF NODES, SECOND IS
C...NO. OF SIDES). IT MUST TALLY WITH COMPUTED GO TO IN SFR.
C
C
      REWIND 17
C
C...FIND NO. OF DIRECT STRESS COMPONENTS NDSTR
      NDSTR = 2
      IF (MSTYP.GE.2) NDSTR = 3
C
      DO 80 L=1,MELEM
      MNODE = LNODN(L)
      NSIDE = LSIDN(L)
      NODSID = MNODE*10 + NSIDE
      DO 5 LTYPE=1,9
      IF(NODSID.EQ.LTYPS(LTYPE)) GO TO 10
    5 CONTINUE
C...IF DO 5 LOOP COMPLETED ELEMENT TYPE IS IMPERMISSIBLE
      GO TO 300
   10 CONTINUE
```

```
C...FIND LSORT: 0=STRUCTURE, 1=SOIL ELEMENT
      LSORT = LMTYP(L) - 1
C
C...FIND NEVAB.(IT IS NOT ALWAYS NDOFN*NNODE SINCE NDOFN CAN VARY)
      NEVAB = 0
      DO 15 N=1,NNODE
      NN = LNODS(L,N)
      NDOFN = NPNDF(NN)
      NEVAB = NEVAB + NDOFN
C...FOR STRUCT. ELTS, IGNORE PORE-PRESSURE D.O.F.S
      IF (LSORT.EQ.0.AND.NDOFN.GT.NDSTR) NEVAB=NEVAB-NDOFN+NDSTR
C...WHILE IN THIS LOOP PUT CO-ORDS IN LOCAL ARRAYS
      DO 14 M=1,3
   14 ENCOD(M,N) = COORD(NN,M)
   15 CONTINUE
      IF (LSORT.EQ.0) GO TO 12
C...DISCOUNT PORE-PRESS. DOF'S FROM NEVAB
C...LET LPORN(N)=1 IF NODE N HAS A PORE-PRESS. DOF.
      NSVAB = NEVAB
      NPORN = 0
      DO 7 N=1,NNODE
      NN = LNODS(L,N)
      NDOFN = NPNDF(NN)
      LPORN(N) = 0
      IF (NDOFN.EQ.NDSTR) GO TO 7
      NPORN = NPORN + 1
      LPORN(N) = 1
      DO 6 M=1,3
    6 ENCDP(M,NPORN) = ENCOD(M,N)
      NEVAB = NEVAB - 1
    7 CONTINUE
    8 CONTINUE
C...FIND ELT TYPE FOR PORE-PRESS NODES: LTYPP
      NODSDP = NPORN*10 + NSIDE
      DO 9 LTYPP=1,9
      IF (NODSDP.EQ.LTYPS(LTYPP)) GO TO 11
    9 CONTINUE
      GO TO 400
   11 CONTINUE
   12 CONTINUE
C
C
C...EXTRACT MAT. PROPS. INTERP. MIDSIDE NODE VALUE IF NEC.
      DO 19 M=1,NPROP
      SUM = 0.D0
      DO 17 N=1,NNODE
      NSET = LPROS(L,N)
      IF(NSET.GT.0) GO TO 16
      VALUE(N) = 0.D0
      GO TO 17
   16 VALUE(N) = PROPS(NSET,M)
      SUM = SUM + DABS(VALUE(N))
   17 CONTINUE
      IF(SUM.EQ.0.D0) CALL MSNVAL(NNODE,NSIDE,VALUE)
      DO 18 N=1,NNODE
   18 ENPRO(M,N) = VALUE(N)
   19 CONTINUE
C
C...ZERO STIFFNESS MATRIX
      DO 20 N=1,NEVAB
```

```
      DO 21 M=1,MEVAB
      ESTIF(M,N) = 0.0D0
      XMATX(M,N) = 0.0D0
      YMATX(M,N) = 0.0D0
   21 CONTINUE
      DO 20 M=1,MNODE
      UMATX(M,N) = 0.0D0
   20 CONTINUE
      DO 22 M=1,MNODE
      DO 22 N=1,MNODE
      MMATX(M,N) = 0.0D0
      SMATX(M,N) = 0.0D0
   22 CONTINUE
C
C
C
      CALL GAUSS(NGAUS,LTYPE,NGTOT,GPWTS,GPLOC)
C
C...ENTER SINGLE LOOP THROUGH ALL GAUSS POINTS
      DO 65 IJ=1,NGTOT
      S = GPLOC(1,IJ)
      T = GPLOC(2,IJ)
      V = GPLOC(3,IJ)
C
C...EVALUATE SHAPE FUNCTIONS AND DERIVS, THEN JACOBIAN AND CART. DERIVS
      CALL SFR(S,T,V,LTYPE,MNODE,SHAPE,DERIV)
      DO 25 M=1,3
      SUM = 0.D0
      DO 24 N=1,MNODE
   24 SUM = SUM + ENCOD(M,N)*SHAPE(N)
   25 GPCOD(M) = SUM
      THICK = 1.
      IF(NDIME.LE.2) THICK = GPCOD(3)
      CALL JACOB(MNODE,MNODE,LTYPE,NDIME,L,ENCOD,SHAPE,DERIV,CARTD,
     1DJACB)
      DVOLU = DJACB*GPWTS(IJ)*THICK
C
C...EVALUATE B MATRIX, R IS RADIUS NEEDED FOR AXI-SYM CASE
      IF (LTYPE.GT.2) GO TO 30
      CALL BARB(MEVAB,MNODE,ENCOD,SHAPE,DERIV,CARTD,DJACB,SCONV,
     1BMATX)
      GO TO 32
   30 R = GPCOD(1)
      IF(R.LT.SMALL) R=SMALL
      CALL BMAT(MPOIN,MELEM,MNODE,MEVAB,NNODE,MSTYP,L,LNODS,NPNDF
     1,R,SHAPE,CARTD,SCONV,BMATX)
C
C...EXTRACT YOUNGS MOD. AND POISSONS RATIO, THEN FILL IN D MATRIX
C...ALSO FIND PERMEABILITIES, POROSITY & PWBM
C...FOR SOIL, YM=ENPRO(1,N), FOR STRUCT ELT YM=ENPRO(3,N)
   32 PERM(1) = 0.0D0
      PERM(2) = 0.0D0
      PERM(3) = 0.0D0
      POROS = 0.0D0
      YM = 0.D0
      PR = 0.D0
      DO 35 N=1,MNODE
      YM = YM + ENPRO(1,N)*SHAPE(N)
      PR = PR + ENPRO(2,N)*SHAPE(N)
      DO 26 I=1,NDSTR
```

```
   20 PERM(I) = PERM(I) + ENPRO(I+5,N)*SHAPE(N)
      POROS = POROS + ENPRO(5,N)*SHAPE(N)
   35 CONTINUE
      IF (LSORT.GE.2) YM = 1.0D0
C
      CALL MOD1(MSTYP,NSTRE,YM,PR,DMATX)
C
C...CALC D*B
      DO 40 M=1,NEVAB
      DO 40 I=1,NSTRE
      SUM = 0.D0
      DO 39 J=1,NSTRE
   39 SUM = SUM + DMATX(J,I)*BMATX(J,M)
      DBMAT(I,M) = SUM
   40 CONTINUE
C
C...THE CONSUMMATION OF THE ELEMENT STIFFNESS CALCULATION
      DO 50 M=1,NEVAB
      DO 50 N=M,NEVAB
      SUM = 0.D0
      DO 49 I=1,NSTRE
   49 SUM = SUM + BMATX(I,M)*DBMAT(I,N)
   50 ESTIF(M,N) = ESTIF(M,N) + SUM*DVOLU
      IF (LSORT.EQ.0) GO TO 65
C
C...NOW FORM PORE-PRESS. SUBMATRICES. FIRST EVALUATE SHAPE &
C...CARTD FOR PORE-PRESS. NODES, IF NPORN.NE.NNODE.
      IF (NPORN.EQ.NNODE) GO TO 52
      CALL SFR(S,T,V,LTYPP,MNODE,SHAPE,DERIV)
      CALL JACOB(MNODE,NPORN,LTYPP,NDIME,L,ENCOP,SHAPE,DERIV,CARTD,
     1DJACB)
   52 CONTINUE
C...FORM U MATRIX (L MATRIX IN STANDARD NOTATION)
      DO 55 J=1,NPORN
      DO 55 I=1,NEVAB
      SUML = 0.0D0
      DO  54 K=1,NDSTR
   54 SUML = SUML + BMATX(K,I)*SHAPE(J)
   55 UMATX(J,I) = UMATX(J,I) + SUML*DVOLU
C...FORM H & S MATRICES
      DO 60 J=1,NPORN
      DO 60 I=1,NPORN
      SUMH = 0.0D0
      SUMS = 0.0D0
      DO 59 K=1,NDSTR
   59 SUMH = SUMH + PERM(K)*CARTD(K,I)*CARTD(K,J)
      SUMS = SUMS + SHAPE(I)*SHAPE(J)*CMPRS
      HMATX(I,J) = HMATX(I,J) + SUMH*DVOLU
      SMATX(I,J) = SMATX(I,J) + SUMS*DVOLU
   60 CONTINUE
   65 CONTINUE
C
C...FILL IN LOWER TRIANGLE OF STIFFNESS MATRIX
      DO 70 M=2,NEVAB
      M1 = M-1
      DO 69 N=1,M1
   69 ESTIF(M,N) = ESTIF(N,M)
   70 CONTINUE
C
      IF (LSORT.NE.0) GO TO 100
```

```
C
C... WRITE TO DISC FILE FOR STRUCTURE ELT
      WRITE (17) ESTIF
      WRITE (17) YMATX
      GO TO 80
C
  100 CONTINUE
C
C...FORM X & Y MATRICES FROM ESTIF,U,N,S, ARRANGING COLUMNS
C...SO THAT PRESSURE AT NODE N COMES AS LAST D.O.F. AT NODE N
      JCOL = 0
      NP = 0
      DO 155 N=1,NNODE
      NN = LNODS(L,N)
      NDOFN = NPNDF(NN)
      JCOL1 = JCOL + 1
      JEND = JCOL + NDOFN - LPORN(N)
      DO 140 J=JCOL1,JEND
      JJNP = J - NP
      DO 135 I=1,NEVAB
      XMATX(I,J) = ESTIF(I,JJNP)
  135 CONTINUE
      NEVB1 = NEVAB + 1
      DO 140 I=NEVB1,NSVAB
      IINVB = I - NEVAB
      YMATX(I,J) = -SCONV*UMATX(IINVB,JJNP)*PSCAL
  140 CONTINUE
      JCOL = JEND
      IF (LPORN(N).EQ.0) GO TO 155
      NP = NP + 1
      JCOL = JCOL + 1
      DO 145 I=1,NEVAB
      XMATX(I,JCOL) = -SCONV*UMATX(NP,I)*PSCAL
  145 CONTINUE
      DO 150 I=NEVB1,NSVAB
      IINVB = I - NEVAB
      XMATX(I,JCOL) = -HMATX(IINVB,NP)*PSCAL*PSCAL
      YMATX(I,JCOL) = -SCONV*SMATX(IINVB,NP)*PSCAL*PSCAL
  150 CONTINUE
  155 CONTINUE
C...NOW SWAP ROWS AROUND CORRESPONDINGLY
      NP = 0
      IROW = 0
      DO 170 N=1,NNODE
      NN = LNODS(L,N)
      NDOFN = NPNDF(NN)
      IROW = IROW + NDOFN
      IF (LPORN(N).EQ.0) GO TO 170
      IROW = IROW - 1
      NP = NP + 1
      DO 165 J=1,NSVAB
      NVBNP = NEVAB + NP
      TEMPX = XMATX(NVBNP,J)
      TEMPY = YMATX(NVBNP,J)
      I = NEVAB + NP - 1
  160 CONTINUE
      XMATX(I+1,J) = XMATX(I,J)
      YMATX(I+1,J) = YMATX(I,J)
      I = I - 1
      IF (I.GE.IROW+1) GO TO 160
```

```
      XMATX(IROW+1,J) = TEMPX
      YMATX(IROW+1,J) = TEMPY
  165 CONTINUE
      IROW = IROW + 1
  170 CONTINUE
C
C...WRITE TO DISC FILE
      WRITE (17) XMATX
      WRITE (17) YMATX
   80 CONTINUE
      RETURN
C
C...ERROR MESSAGE
  300 WRITE(6,700) L,NNODE,NSIDE
      STOP
  400 WRITE (6,800) L,NPORN,NSIDE
      STOP
  700 FORMAT(/38H STOPPED IN STIFF. CANT HANDLE ELEMENT,I4, 8H. IT HAS,
     1 I3,10H NODES AND,I3,6H SIDES)
  800 FORMAT(/38H STOPPED IN STIFF. CANT HANDLE ELEMENT,I4,
     18H. IT HAS,I3,24H PORE-PRESSURE NODES AND,I3,6H SIDES)
      END
C
C
C
      SUBROUTINE LOAD
     1(NTOTV,NPOIN,NELEM,NFIXV,NEVAB,NNODE,NSETS,NCOMP,
     2NPOIN,NELEM,NFIXV,NSDIS,NDIME,NGAUS,NSTYP,NODEM,NBSET,NPSET,
     3LNODS,LNODN,LSIDN,NPNDF,NFVAR,LBOPS,BFOPS,SCONV,
     4COORD,ENCOD,ENBOD,SHAPE,DERIV,CARTD,ASLOD,SPDIS,NTIME,PSCAL)
C
C...READS NODAL LOADS AND SPEC. DISPLACEMENTS, BODY FORCES, AND SURFACE
C...PRESSURES. CONVERTS ALL LOADS TO NODAL FORCES WHICH ARE ASSEMBLED IN
C.. ASLOD. SPEC. DISP. ARE STORED IN SPDIS WHICH INCLUDES PREVIOUS NFIXV
C...FIXED VARIABLES. NSDIS IS THE TOTAL LENGTH OF THIS ARRAY.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION LNODS(NELEM,NNODE),LNODN(NELEM),LSIDN(NELEM)
     1,NPNDF(NPOIN),NFVAR(NFIXV),LBOPS(NELEM,NNODE),BFOPS(NSETS,NCOMP)
     2,COORD(NPOIN,3),ENCOD(3,NNODE),ENBOD(NCOMP,NNODE)
     3,SHAPE(NNODE),DERIV(3,NNODE),CARTD(3,NNODE),GPLOC(3,8),GPWTS(8)
     4,ASLOD(NTOTV),SPDIS(NFIXV),PRESS(3,8),GPRES(3),XJACM(2,3),XJ(3)
     5,FCOMP(3),KARD(20),CARD(20),LTYPS(9),NPALS(8),NASET(8)
C
      DATA IBRAC,JBRAC/1H(,1H)/
      DATA LTYPS/21,31,33,44,63,64,84,86,206/
C...SEE NOTE IN STIFF
C
      NTOTV = NPOSN(NPOIN,NPNDF(NPOIN),NPNDF)
C...NPOSN IS A FUNCTION SUBROUTINE WHICH FINDS VARIABLE ADDRESS
      ISTOP = 0
      DO 5 I=1,NTOTV
    5 ASLOD(I) = 0.D0
      IF (NTIME.GT.0) GO TO 6
      DO 7 I=1,NFIXV
    7 SPDIS(I) = 0.0D0
    6 CONTINUE
      NSDIS = NFIXV
C...FIND NDSTR; IF DOF.GT.NDSTR IT IS A PORE-PRESSURE D.O.F.
      NDSTR = 2
```

```
      IF (NSTYP.GE.2) NDSTR = 3
C
C...INPUT NODAL FORCES AND DISPLACEMENTS (CARDS LD)
    9 WRITE (6,610)
   10 READ(5,510)LCARD,LD,N,(KARD(I),CARD(I),I=1,7)
      IF(LD.EQ.1.OR.LD.EQ.2) GO TO 11
      IF(N.EQ.0) GO TO 21
      ISTOP = ISTOP + 1
      WRITE(6,710) N
      GO TO 21
   11 IF(N.GT.0.AND.N.LE.NPOIN) GO TO 12
      ISTOP = ISTOP + 1
      WRITE(6,711) N
      GO TO 21
   12 DO 20 I=1,7
      K = KARD(I)
      IF(K.LE.0) GO TO 20
      NN = NPOSN(N,K,NPNDF)
      GO TO (15,16),LD
   15 ASLOD(NN) = CARD(I)
      GO TO 20
C...DISP IS SPEC. IS IT A FIXED D.OF F.? IF YES IGNORE.
   16 DO 17 K=1,NFIXV
      IF(NFVAR(K).EQ.NN) GO TO 19
   17 CONTINUE
      NSDIS = NSDIS + 1
      IF(NSDIS.LE.NFIXV) GO TO 18
      ISTOP = ISTOP + 1
      WRITE(6,712)
      GO TO 21
   18 SPDIS(NSDIS) = CARD(I)
      NFVAR(NSDIS) = NN
      GO TO 20
   19 WRITE(6,713) N,K
   20 CONTINUE
   21 IF(LCARD.EQ.0) GO TO 10
C
C...PRINT APPLIED NODAL FORCES AND SPECIFIED DISPLACEMENTS
      NONE = 0
      NN = 0
      DO 30 N=1,NPOIN
      NDOFN = NPNDF(N)
      DO 29 K=1,NDOFN
      NN = NN+1
      F = ASLOD(NN)
      IF(NSDIS.EQ.NFIXV) GO TO 24
      NFV1 = NFIXV + 1
      DO 23 K=NFV1,NSDIS
      IF(NFVAR(K).EQ.NN) GO TO 25
   23 CONTINUE
C...NO SPEC. DISP. (D=0)
   24 IF(F.EQ.0.D0) GO TO 29
      WRITE(6,612) N,K,F
      GO TO 27
C...THERE IS A SPEC. DISP. AND MAY ALSO BE A FORCE
   25 D = SPDIS(K)
      IF (K.GT.NDSTR) SPDIS(K) = SPDIS(K)/PSCAL
      IF(F.EQ.0.D0) GO TO 26
      WRITE(6,613) N,K,F,D
      GO TO 27
```

```
   26 WRITE(6,611) N,M,D
   27 NONE = 1
   29 CONTINUE
   30 CONTINUE
C
C
C...INPUT BODY FORCE SETS AND THEIR DISTRIBUTION (B1,B2)
      IF(NBSET.EQ.0) GO TO 160
      NBCOM = NDIME
      CALL SETVAL(NELEM,NNODE,NSETS,NCOMP,NPOIN,NELEM,NODEM,ISTOP,
     1NBSET,NBCOM,LNODS,LNODN,LBOPS,BFOPS,0,
     25HAPP.B,5HODY F,5HORCES,5H (B1),5H (B2))
C
C...START SECTION TO CONVERT BODY FORCES TO NODAL LOADS
      WRITE(6,650)
      DO 150 L=1,NELEM
      NNODE = LNODN(L)
C...CHECK FOR EXISTANCE OF BODY FORCE
      NON = 0
      DO 55 N=1,NNODE
   55 NON = NON + LBOPS(L,N)
      IF(NON.EQ.0) GO TO 150
C...ELEMENT IS LOADED, COMPUTE MIDSIDE VALUE IF NEC.
      NSIDE = LSIDN(L)
C...PRINT EL. AND NODE NO. BEFORE PRINTING B.F. VALUES.
      NOD = NNODE
      IF(NOD.GT.8) NOD=8
      WRITE(6,651)L,(LNODS(L,N),N=1,NOD)
      IF(NNODE.GT.8)WRITE(6,652)(LNODS(L,N),N=9,NNODE)
      DO 75 M=1,NBCOM
      SUM = 0.D0
      DO 65 N=1,NNODE
      ENBOD(M,N) = 0.D0
      NSET = LBOPS(L,N)
      IF(NSET.GT.0) GO TO 60
      CARD(N) = 0.D0
      GO TO 65
   60 CARD(N) = BFOPS(NSET,M)
      SUM = SUM + CARD(N)
   65 CONTINUE
      IF(SUM.EQ.0.D0) GO TO 75
      CALL MSNVAL(NNODE,NSIDE,CARD)
      DO 70 N=1,NNODE
   70 ENBOD(M,N) = CARD(N)
      NOD = NNODE
      IF(NOD.GT.8) NOD=8
      WRITE(6,653)M,(ENBOD(M,N),N=1,NOD)
      IF(NNODE.GT.8) WRITE(6,654)(ENBOD(M,N),N=9,NNODE)
   75 CONTINUE
C...TRANSFER CO-ORDS TO ELEMENT ARRAY
      DO 85 N=1,NNODE
      NP = LNODS(L,N)
      DO 80 M=1,3
   80 ENCOD(M,N) = COORD(NP,M)
   85 CONTINUE
C...OBTAIN LTYPE - IT IS NEEDED IN GAUSS AND SFR
      NODSID = NNODE*10 + NSIDE
      DO 90 LTYPE=1,9
      IF(NODSID.EQ.LTYPS(LTYPE)) GO TO 95
   90 CONTINUE
```

```
      ISTOP = ISTOP+1
      WRITE(6,735) L,NNODE,NSIDE
      GO TO 150
C...ALL SET FOR GAUSS LOOP
   95 CALL GAUSS(NGAUS,LTYPE,NGTOT,GPWTS,GPLOC)
      IF (LTYPE.LE.2) GO TO 150
      DO 120 IJ=1,NGTOT
      S = GPLOC(1,IJ)
      T = GPLOC(2,IJ)
      V = GPLOC(3,IJ)
      CALL SFR(S,T,V,LTYPE,NNODE,SHAPE,DERIV)
      THICK = 1.
      IF(NDIME.EQ.3) GO TO 105
C...EVALUATE THICKNESS AT GAUSS POINTS, 2D AND AXI-SYM CASES
      THICK = 0.D0
      DO 100 N=1,NNODE
  100 THICK = THICK + ENCOD(3,N)*SHAPE(N)
  105 CALL JACOB(NNODE,NNODE,LTYPE,NDIME,L,ENCOD,SHAPE,DERIV,CARTD,
     1DJACB)
      DVOLU = DJACB*GPWTS(IJ)*THICK
C...ACCUMULATE EQUIV. NODAL FORCES IN ASLOD
      DO 115 N=1,NNODE
      NP = LNODS(L,N)
      NIK = NPOSN(NP,0,NPNDF)
      DO 110 M=1,NDIME
      MM = NIK + M
  110 ASLOD(MM) = ASLOD(MM) + ENBOD(M,N)*SHAPE(N)*DVOLU
  115 CONTINUE
  120 CONTINUE
  150 CONTINUE
C
C
C...INPUT SURFACE PRESSURE SETS AND STORE THEM IN BFOPS (P1)
  160 IF(NPSET.EQ.0) GO TO 350
      NPCOM = 2
      IF(NDIME.EQ.3) NPCOM=1
C...WE LIMIT 3D COMP. TO NORMAL PRESSURE ONLY
      CALL SETVAL(NELEM,NNODE,NSETS,NCOMP,NPOIN,NELEM,NODEM,ISTOP,
     1NPSET,NPCOM,LNODS,LNODM,LBOPS,BFOPS,1.
     25HSURF.,5H TRAC,5HTIONS,5H (P1),5H     )
C
C...SPECIFY DISTRIBUTION OF PRESSURE SETS (P2)
C...(WE CANT USE SETVAL FOR THIS SINCE FORMATS DIFFERENT)
      WRITE(6,670)
      NONE = 0
      NEDGE = 1
      IF(NDIME.EQ.3) NEDGE=4
      NJDIM = NDIME - 1
  170 READ(5,571)LCARD,NNAS,L,(NPALS(N),NASET(N),N=1,8)
C...NOTE: 1 DATA CARD FOR EACH LOADED SIDE (OR FACE IN 3D)
      IF(NDIME.EQ.1) GO TO 300
      WRITE(6,671) L,(NPALS(N),IBRAC,NASET(N),JBRAC,N=1,NNAS)
      NONE = 1
      IF(NNAS.GE.2.AND.NNAS.LE.8) GO TO 175
      ISTOP = ISTOP+1
      WRITE(6,770) NNAS
      GO TO 300
  175 IF(L.LE.NELEM) GO TO 180
      ISTOP = ISTOP+1
      WRITE(6,733) L
```

```
      GO TO 300
  180 NNODE = LNODN(L)
C...CHECK SPEC. NODES ARE IN SEQUENCE ALONG SIDE
C...AND PUT MIDSIDE NODE NUMBERS IN NPALS() IF NEC.
      CALL PCHECK(NELEM,MNODE,NNODE,NNAS,L,LNODS,NPALS,ISTOP)
C...CHECK SET NO.S
      DO 195 NAS=1,NNAS
      NSET = NASET(NAS)
      IF(NSET.GE.0.AND.NSET.LE.NPSET) GO TO 195
      ISTOP = ISTOP+1
      WRITE(6,780) NSET
      GO TO 300
  195 CONTINUE
C...PUT TRACTION VALUES IN PRESS()
      DO 215 N=1,NPCOM
      SUM = 0.D0
      DO 205 N=1,NNAS
      PRESS(N,N) = 0.D0
      NSET = NASET(N)
      IF(NSET.GT.0) GO TO 200
      CARD(N) = 0.D0
      GO TO 205
  200 CARD(N) = BFOPS(NSET,N)
      SUM = SUM + CARD(N)
  205 CONTINUE
      IF(SUM.EQ.0.D0) GO TO 215
      CALL NSHVAL(NNAS,NEDGE,CARD)
      DO 210 N=1,NNAS
  210 PRESS(N,N) = CARD(N)
      WRITE(6,672) N,(PRESS(N,N),N=1,NNAS)
  215 CONTINUE
C...EXTRACT NODAL CO-ORDS
      DO 225 N=1,NNAS
      NP = NPALS(N)
      DO 220 N=1,3
  220 ENCOD(N,N) = COORD(NP,N)
  225 CONTINUE
C...THE VARIOUS CHECKS OVER WE CAN NOW ENTER GAUSS LOOP ALONG SIDE
C...NOTE SIDE IS A LINE EL. IN 2D AND A PLANE EL. IN 3D
C...OBTAIN SIDE TYPE (LSTYP). BY COINCIDENCE LSTYP = NNAS-1 EXCEPT
C...FOR 8 NODE BRICK
      LSTYP = NNAS - 1
      IF(NNAS.EQ.4) LSTYP=4
      CALL GAUSS(NGAUS,LSTYP,NGTOT,GPWTS,GPLOC)
      DO 290 IJ=1,NGTOT
      S = GPLOC(1,IJ)
      T = GPLOC(2,IJ)
      CALL SFR(S,T,0.D0,LSTYP,MNODE,SHAPE,DERIV)
C...COMPUTE APPROPRIATE JACOBIAN COMPS.
      DO 245 J=1,NDIME
      DO 245 I=1,NJDIM
      SUM = 0.D0
      DO 240 N=1,NNAS
  240 SUM = SUM + DERIV(I,N)*ENCOD(J,N)
      XJACM(I,J) = SUM
  245 CONTINUE
C...PUT PRESSURE COMP. IN LOCAL ARRAY
      DO 255 N=1,NPCOM
      SUM = 0.D0
      DO 250 N=1,NNAS
```

```
  250 SUM = SUM + PRESS(M,N)*SHAPE(N)
  255 GPRES(M) = SUM
      GO TO (260,270),NJDIM
C...2D CASE (LOAD APPLIED TO EDGE)
  260 THICK = 0.D0
      DO 265 N=1,NNAS
  265 THICK = THICK + ENCOD(3,N)*SHAPE(N)
      DLONG = THICK*GPWTS(IJ)*SCONV
      FCOMP(1) = -(XJACM(1,1)*GPRES(2) - XJACM(1,2)*GPRES(1))*DLONG
      FCOMP(2) = -(XJACM(1,1)*GPRES(1) + XJACM(1,2)*GPRES(2))*DLONG
      GO TO 280
C...3D CASE (LOAD APPLIED TO FACE)
  270 XJ(1) =  XJACM(1,2)*XJACM(2,3) - XJACM(1,3)*XJACM(2,2)
      XJ(2) =  XJACM(1,3)*XJACM(2,1) - XJACM(1,1)*XJACM(2,3)
      XJ(3) =  XJACM(1,1)*XJACM(2,2) - XJACM(1,2)*XJACM(2,1)
      DPA = GPWTS(IJ)*GPRES(1)*SCONV
      DO 275 M=1,3
  275 FCOMP(M) = XJ(M)*DPA
C...ACCUMULATE EQUIV. NODAL FORCES IN ASLOD
  280 DO 285 N=1,NNAS
      NP = NPNLS(N)
      NIK = NPOSN(NP,0,NPNDF)
      DO 282 M=1,NDIME
      MM = NIK + M
  282 ASLOD(MM) = ASLOD(MM) + FCOMP(M)*SHAPE(N)
  285 CONTINUE
  290 CONTINUE
  300 IF(LCARD.EQ.0) GO TO 170
C
  350 IF(ISTOP.EQ.0) RETURN
      WRITE(6,800) ISTOP
      STOP
C
C...INPUT FORMATS
  510 FORMAT(I1,I4,I5,7(I1,D9.3))
  520 FORMAT(I1,I4,3X,6I2,6D10.3)
  571 FORMAT(I1,I4,I5,8(I3,I2))
C
C...OUTPUT FORMATS
  610 FORMAT(//45H APPLIED NODAL FORCES AND SPEC. DISPLACEMENTS,
     1/1X,44(1H-)//3X,4HNODE,3X,6HD OF F,3X,5HFORCE,6X,11HSPEC. DISP.)
  611 FORMAT(I6,I8,14X,D13.5)
  612 FORMAT(I6,I8, 1X,D13.5)
  613 FORMAT(I6,I8, 1X,2D13.5)
  650 FORMAT(/24H BODY FORCE NODAL VALUES)
  651 FORMAT(3X,7HELEMENT,I4,17X,23HN O D E   N U M B E R S,
     1/4X,5HCOMP.,I10,7I13)
  652 FORMAT(8X,8I13)
  653 FORMAT(I7,4X,8D13.5)
  654 FORMAT(11X,8D13.5)
  670 FORMAT(//53H SURFACE TRACTION SET NO. CONNECTIONS AND VALUES (P2),
     1/1X,52(1H-))
  671 FORMAT( 4H EL.,I3,11H, NODE(SET),8(3X,I3,A1,I2,A1,3X))
  672 FORMAT(5X, 9HCOMP. NO.,I2,1X,8D13.5)
C
C...ERROR MESSAGE FORMATS
  710 FORMAT(45H *** FATAL ERROR, LD NEITHER 1 NOR 2 FOR NODE,I4)
  711 FORMAT(24H *** FATAL ERROR, NODE =,I4,24H OUTSIDE PERMITTED LIMIT)
  712 FORMAT(38H *** FATAL ERROR, SPDIS ARRAY EXCEEDED)
  713 FORMAT(46H *** WARNING, ATTEMPT TO SPECIFY DISP. AT NODE
```

```
     1,I4,' 8H D.OF F.,I2,30H, THIS D.OF F. HAS BEEN FIXED.)
  732 FORMAT(24H *** FATAL ERROR, EL. NO.I5,14H IMPERMISSIBLE)
  735 FORMAT(37H *** FATAL ERROR, CANT HANDLE ELEMENT,I4,10H WHICH HAS,
     1I3,10H NODES AND,I2,6H SIDES)
  770 FORMAT(24H *** FATAL ERROR, NNAS =,I5,14H IMPERMISSIBLE)
  775 FORMAT(28H *** FATAL ERROR, SPEC. NODE,I5,36H DOES NOT MATCH NODES
     1 AROUND ELEMENT,I4)
  780 FORMAT(24H *** FATAL ERROR, NSET =,I4,28H IS OUTSIDE RANGE OF P1 D
     1ATA)
  800 FORMAT(///I5,45H FATAL ERRORS IN LOAD. BETTER LUCK NEXT TIME.)
      END
C
C
C
      SUBROUTINE FRONT(MTOTV,MPOIN,MELEM,MFIXV,MEVAB,MNODE,MFRON,MSTIF,
     1MPOIN,MELEM,MSDIS,LNODS,LNODN,MPNDF,MFVAR,ASLOD,ASDIS,SPDIS,
     2VECRV,ESTIF,GSTIF,GLOAD,EQUAT,NACVA,LOCEL,NDEST,PIVAL,IRSOL,ASLD1,
     3XMATX,YMATX,THETA,MTIME,LMTYP,MSTYP,NGAUS,SHAPE,DERIV,
     4DTIME,LPORN,PSTSL,ASRHS,LPROS,PROPS,MSETS,MCOMP,REACT,LSIDN)
C
C...SOLVES FOR LOAD VECTOR ASLOD AND RETURNS WITH DISPLACEMENTS IN ASDIS
C...LOADS UNALTERED IN ASLOD EXCEPT FOR ADDITION OF REACTIONS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION LNODS(MELEM,MNODE),LNODN(MELEM),MPNDF(MPOIN)
     1,MFVAR(MFIXV),ASLOD(MTOTV),ASDIS(MTOTV),SPDIS(MFIXV)
     2,VECRV(MFRON),ESTIF(MEVAB,MEVAB),GSTIF(MSTIF),GLOAD(MFRON)
     3,EQUAT(MFRON),NACVA(MFRON),LOCEL(MEVAB),NDEST(MEVAB)
     4,ASLD1(MTOTV),XMATX(MEVAB,MEVAB),YMATX(MEVAB,MEVAB),LMTYP(MELEM)
     5,LPORN(MNODE),PSTSL(MELEM,4),ASRHS(MFIXV),LPROS(MELEM,MNODE)
     6,PROPS(MSETS,MCOMP),LTYPS(9),GPLOC(3,8),SHAPE(MNODE)
     7,DERIV(3,MNODE),GPWTS(8),LSIDN(MELEM)
C...VECRV...NDEST INTERNAL USE ONLY
C
      DATA LTYPS/21,31,33,44,63,64,84,86,206/
C
C...FUNCTION STATEMENT
      NFUNC(I,J) = (J*J - J)/2 + I
      IF(NFUNC(MFRON,MFRON).GT.MSTIF) GO TO 500
C
C
      THT = THETA/(1.D0 - THETA)
C...FIND NO. OF DISPLACEMENT D.O.F.S, NDISP
      NDISP = 2
      IF (MSTYP.GE.2) NDISP = 3
C
C...CHANGE SIGN OF LAST APPEARANCE OF EACH NODE
      DO 40 NP=1,NPOIN
      KLAST = 0
      DO 30 L=1,NELEM
      NNODE = LNODN(L)
      DO 20 N=1,NNODE
      IF(LNODS(L,N).NE.NP) GO TO 20
      KLAST = L
      NLAST = N
   20 CONTINUE
   30 CONTINUE
      IF(KLAST.NE.0) LNODS(KLAST,NLAST) = -NP
   40 CONTINUE
C
```

```
C...ZERO ARRAYS
      DO 50 I=1,NSTIF
   50 GSTIF(I) = 0.D0
      DO 60 I=1,NFRON
      GLOAD(I) = 0.D0
      EQUAT(I) = 0.D0
      VECRV(I) = 0.D0
   60 NACVA(I) = 0
      DO 61 I=1,NFIXV
   61 ASRHS(I) = 0.0D0
      NFMAX = 0
C
C...PUT ASLD1 INTO ASLOD
      DO 65 NV=1,NTOTV
      ASLOD(NV) = ASLD1(NV)
   65 CONTINUE
C
      REWIND 17
      REWIND 18
      IF (IRSOL.NE.0) REWIND 19
C
C
C...ENTER MAIN ASSEMBLY - REDUCTION LOOP
      NFRON = 1
      KELVA = 0
C
  100 DO 380 L=1,NELEM
      NNODE = LNODN(L)
C...SET UP LOCEL TO RELATE ELEMENT VARIABLE POSITIONS TO OVERALL POSNS
      NEVAB = 0
      DO 170 N=1,NNODE
      NP = LNODS(L,N)
      IS = 1
      IF(NP.GT.0) GO TO 160
      IS = -1
      NP = -NP
  160 NDOFN = NPNDF(NP)
      NIK = NPOSN(NP,0,NPNDF)
      DO 165 I=1,NDOFN
C...FOR STRUCT. ELTS, DON'T COUNT PORE-PRESSURE D.O.F.S
      IF (LMTYP(L).NE.1) GO TO 164
      IF (NSTYP.NE.3.AND.I.GT.2) GO TO 165
      IF (NSTYP.EQ.3.AND.I.GT.3) GO TO 165
  164 NEVAB = NEVAB + 1
      LOCEL(NEVAB) = (NIK+I)*IS
  165 CONTINUE
      LNODS(L,N) = NP
  170 CONTINUE
C...THE ROLE OF LNODS FOR LOCATING LAST NODE APPEARANCE HAS BEEN TRANS-
C...FERRED TO LOCEL. HENCEFORTH WE FORGET D.OF F. AT NODE AND CONSIDER
C...ONLY THE VARIABLE POSITION (IE EQN NO.)
C
C...LOOK FOR EXISTING DESTINATIONS
      KEVAB = 0
      DO 210 IE=1,NEVAB
      NIKNO = IABS(LOCEL(IE))
      KEXIS = 0
      DO 180 I=1,NFRON
      IF(NIKNO.NE.NACVA(I)) GO TO 180
      KEVAB = KEVAB+1
```

```
      KEXIS = 1
      NDEST(KEVAB) = I
  180 CONTINUE
      IF(KEXIS.EQ.1) GO TO 210
C...ALLOCATE VARIABLE A SPACE IN NACVA
      DO 190 I=1,NFRON
      IF(NACVA(I).NE.0) GO TO 190
      NACVA(I) = NIKNO
      KEVAB = KEVAB+1
      NDEST(KEVAB) = I
      GO TO 200
  190 CONTINUE
      GO TO 502
C...EXPAND FRONT WIDTH AS REQUIRED
  200 NF = NDEST(KEVAB)
      IF(NF.GT.NFRON) NFRON=NF
      IF(NFRON.GT.NFMAX) NFMAX = NFRON
  210 CONTINUE
C
C...CHECK NSTIF DIMENSION SUFFICIENT
      NSTIF = NFRON*(NFRON+NFRON)
      IF(NSTIF.GT.MSTIF) GO TO 503
C
      IF (IRSOL.NE.0) GO TO 216
C...FORM L.H.S. COMBINED MATRIX
  213 READ (17) XMATX
      READ (17) YMATX
      IF (LMTYP(L).EQ.1) GO TO 90
C
      IF (LMTYP(L).EQ.2) GO TO 80
C...FIND YOUNG'S MOD FOR CURRENT NONLINEAR-LAW ELEMENT
      YM = 0.0D0
      NSIDE = LSIDN(L)
      NODSID = NNODE*10 + NSIDE
      DO 82 LTYPE=1,9
      IF (NODSID.EQ.LTYPS(LTYPE)) GO TO 72
   82 CONTINUE
      GO TO 505
   72 CALL GAUSS(NGAUS,LTYPE,NGTOT,GPWTS,GPLOC)
      DO 77 IJ=1,NGTOT
      S = GPLOC(1,IJ)
      T = GPLOC(2,IJ)
      V = GPLOC(3,IJ)
      CALL SFR(S,T,V,LTYPE,NNODE,SHAPE,DERIV)
      YMGP = 0.0D0
      DO 85 N=1,NNODE
      IF (LMTYP(L).EQ.3) CALL YMMYP(MELEM,MNODE,MSETS,MCOMP,L,N,
     1PSTSL,LPROS,PROPS,REACT,YMN)
      IF (LMTYP(L).EQ.4) CALL YMLIN(MELEM,MNODE,MSETS,MCOMP,L,N,
     1PSTSL,LPROS,PROPS,YMN)
      YMGP = YMGP + YMN*SHAPE(N)
   85 CONTINUE
      YM = YM + YMGP*GPWTS(IJ)
   77 CONTINUE
      NGD = (1 + NSIDE)/2
      AA = DFLOAT(NGAUS**NGD)
      A = 1./AA
      IF (NSIDE.EQ.3) A=2.
      YM = YM*A
   80 CONTINUE
```

```
C
C...CONVERT X TO  X' BY MULT. PORE-PRESS. ROWS & COLS OF X
C...BY (1-THETA)*DT. USE LPORN TO HOLD PORE-PRESS. DOF'S.
      NDOF = 0
      NPDOF = 0
      DO 88 N=1,NNODE
      NP = LNODS(L,N)
      NDOFN = NPNDF(NP)
      NDOF = NDOF + NDOFN
      IF (NDOFN.LE.NDISP) GO TO 88
      NPDOF = NPDOF + 1
      LPORN(NPDOF) = NDOF
   88 CONTINUE
   86 DO 89 I=1,NDOF
      DO 89 J=1,NDOF
      IP = 0
      JP = 0
      DO 87 KOUNT=1,NPDOF
      IPDOF = LPORN(KOUNT)
      IF (I.EQ.IPDOF) IP = 1
      IF (J.EQ.IPDOF) JP = 1
   87 CONTINUE
C...MULT. ELTS OF XMATX BY (1-THETA)*DT
      IF (IP.EQ.1.AND.JP.EQ.1) XMATX(I,J)=XMATX(I,J)*
     1(1.D0-THETA)*DTIME
      IF (LMTYP(L).LT.3) GO TO 89
C...FOR HYPERBLC-LAW ELTS, MULT ELTS OF ESTIF BY YOUNG'S MOD
      IF (IP.EQ.0.AND.JP.EQ.0) XMATX(I,J)=XMATX(I,J)*YM
   89 CONTINUE
C
   90 CONTINUE
C
      IF (NTIME.EQ.0) GO TO 75
      DO 70 IE=1,NEVAB
      NIK = LOCEL(IE)
      NIK = IABS(NIK)
C...ADD COMBINED RHS MATRIX*OLD DISP. TO ASLOD
      ASLE = 0.0D0
      DO 95 KE=1,NEVAB
      NKK = LOCEL(KE)
      NKK = IABS(NKK)
      ASLE = ASLE + (THT*XMATX(IE,KE) - YMATX(IE,KE))
     1*ASDIS(NKK)
   95 CONTINUE
      ASLOD(NIK) =  ASLOD(NIK) - ASLE
      DO 96 ISDIS = 1,NSDIS
   96 IF (NFVAR(ISDIS).EQ.NIK) ASRHS(ISDIS)=ASRHS(ISDIS) + ASLE
   70 CONTINUE
   75 CONTINUE
      DO 215 J=1,NEVAB
      DO 215 I=1,NEVAB
      ESTIF(I,J) = XMATX(I,J) + YMATX(I,J)
  215 CONTINUE
C
C...COMMENCE ASSEMBLY - FIRST LOADS
  216 DO 240 IE=1,NEVAB
      I = NDEST(IE)
      NIK = LOCEL(IE)
      NIKLE = -NIK
      IF (NIK.LT.0) GLOAD(I) = GLOAD(I) + ASLOD(NIKLE)
```

```
      IF(IRSOL.NE.0) GO TO 230
C...AND THEN ELEMENT STIFFNESSES - SKIP IF RESOLUTION
      DO 220 JE=1,IE
      J = NDEST(JE)
      NASH = NFUNC(I,J)
      NISH = NFUNC(J,I)
      STIF = ESTIF(IE,JE)
      IF(J.GE.I) GSTIF(NASH) = GSTIF(NASH) + STIF
      IF(J.LT.I) GSTIF(NISH) = GSTIF(NISH) + STIF
  220 CONTINUE
  230 CONTINUE
  240 CONTINUE
C
C
C...FIND OUT WHICH ELEMENT NODES CAN BE ELIMINATED
      DO 370 IE=1,NEVAB
      NIKNO = -LOCEL(IE)
      IF(NIKNO.LE.0) GO TO 370
C...VARIABLES TO BE ELIMINATED NOW IDENTIFIED - FIND THEIR POSITIONS
      DO 350 I=1,NFRON
      IF(NACVA(I).NE.NIKNO) GO TO 350
C
      IF(IRSOL.NE.0) GO TO 260
C...EXTRACT COEFS OF NEW EQN FOR ELIMINATION (FIRST SOLUTION ONLY)
      DO 250 J=1,NFRON
      IF(I.LT.J) NL = NFUNC(I,J)
      IF(I.GE.J) NL = NFUNC(J,I)
      EQUAT(J) = GSTIF(NL)
      GSTIF(NL) = 0.D0
  250 CONTINUE
C...AND THE CORRESPONDING R.H.S.
  260 EQRHS = GLOAD(I)
      GLOAD(I) = 0.D0
      KELVA = KELVA+1
C
      IF(IRSOL.NE.0) GO TO 270
      II = I
      WRITE (18) EQUAT,EQRHS,II,NIKNO
      GO TO 280
  270 WRITE (19) EQRHS
      READ (18) EQUAT,DUMMY,II,NIKNO
C
C...TAKE DIAGONAL COEF. OF VARIABLE TO BE ELIM. AS PIVOT. CHECK FOR SIZE
  280 PIVOT = EQUAT(II)
      IF (DABS(PIVOT).LT.100.*PIVAL) WRITE (6,650) PIVOT,NIKNO
      IF (NIKNO.EQ.1) PVMAX = DABS(PIVOT)
      IF (NIKNO.EQ.1) PVMIN = DABS(PIVOT)
      IF (DABS(PIVOT).LT.PVMIN) PVMIN = DABS(PIVOT)
      IF (DABS(PIVOT).GT.PVMAX) PVMAX = DABS(PIVOT)
      IF (DABS(PIVOT).LT.PIVAL) GO TO 501
      EQUAT(II) = 0.D0
C
C...ENQUIRE IF VARIABLE NIKNO FREE OR PRESCRIBED
      DO 285 N=1,NSDIS
      IF(NFVAR(N).EQ.NIKNO) GO TO 287
  285 CONTINUE
      GO TO 300
C...NIKNO IS PRESCRIBED
  287 SPD = SPDIS(N)
      DO 290 J=1,NFRON
```

```
  290 GLOAD(J) = GLOAD(J) - SPD*EQUAT(J)
      GO TO 340
C
C...ELIMINATE FREE VARIABLE - FIRST DEAL WITH RHS THEN LHS COEFS.
  300 EQPIV = EQRHS/PIVOT
      DO 330 J=1,NFRON
      EQJ = EQUAT(J)
      GLOAD(J) = GLOAD(J) - EQJ*EQPIV
      IF(IRSOL.NE.0) GO TO 320
      IF(EQJ.EQ.0.D0) GO TO 330
      NL = NFUND(0,J)
      EQJP = EQJ/PIVOT
      DO 310 K=1,J
      KN = K + NL
  310 GSTIF(KN) = GSTIF(KN) - EQJP*EQUAT(K)
  320 CONTINUE
  330 CONTINUE
C
  340 EQUAT(II) = PIVOT
      NACVA(II) = 0
      GO TO 380
  350 CONTINUE
C...ELIMINATION COMPLETE
C
C...REDUCE FRONT WIDTH IF POSSIBLE
  360 IF(NACVA(NFRON).NE.0) GO TO 370
      NFRON = NFRON-1
      IF(NFRON.GT.0) GO TO 360
  370 CONTINUE
  380 CONTINUE
C...ELEMENT LOOP COMPLETED IN FORWARD ELIMINATION
C
C
C
C...START BACK SUBSTITUTION.
      DO 420 IE=1,KELVA
      BACKSPACE 18
      READ (18) EQUAT,EQRHS,I,NIKNO
      BACKSPACE 18
      IF(IRSOL.EQ.0) GO TO 390
C...RESOLUTIONS ONLY
      BACKSPACE 19
      READ (19) EQRHS
      BACKSPACE 19
C
C...CHECK IF DISP. PRESCRIBED
  390 DO 400 N=1,NSDIS
      IF (NFVAR(N).EQ.NIKNO) GO TO 410
  400 CONTINUE
C...DISP. NOT PRESCRIBED - COMPUTE IT
  401 PIVOT = EQUAT(I)
      EQUAT(I) = 0.D0
      DO 405 J=1,NFRON
  405 EQRHS = EQRHS - VECRV(J)*EQUAT(J)
      VECRV(I) = EQRHS/PIVOT
      GO TO 415
C...DISP. IS PRESCRIBED - PUT REACTIONS (WHICH INCL APP. LOADS) IN ASLD1
  410 VECRV(I) = SPDIS(N)
      DO 412 J=1,NFRON
  412 EQRHS = EQRHS - VECRV(J)*EQUAT(J)
```

```
      ASLD1(NIKNO) = -EQRHS + ASRHS(N)
  415 ASDIS(NIKNO) = VECRV(I)
  420 CONTINUE
C
C...ADVISE WHAT MAX FRONT WIDTH WAS
      WRITE(6,600) NFMAX
      WRITE (6,651) PVMIN,PVMAX
      RETURN
C
C
C...ERROR MESSAGES
  500 WRITE(6,700)
      STOP
  501 WRITE(6,701) NIKNO
      STOP
  502 WRITE(6,702) NFRON,L
      STOP
  503 WRITE(6,703) MSTIF,L,NSTIF
      STOP
  505 WRITE (6,705) L
      STOP
  600 FORMAT(/37H FRONT COMPLETED. MAX FRONT WIDTH WAS,I4)
  650 FORMAT( 26H WARNING IN FRONT. PIVOT =,D12.4, 8H FOR EQN,I4,13H IS
     1DANGEROUS)
  651 FORMAT(/19H MIN. PIVOT SIZE = ,D12.4,10X,
     119H MAX. PIVOT SIZE = ,D12.4)
  700 FORMAT(/37H STOPPED AT START OF FRONT. NFRON TOO LARGE IN RELATION
     1 TO MSTIF. ALTER THEM IN MASTER.)
  701 FORMAT(/42H STOPPED IN FRONT, PIVOT TOO SMALL FOR EQN,I4)
  702 FORMAT(/26H STOPPED IN FRONT, NFRON =,I4,21H EXCEEDS MFRON IN EL.
     1,I4)
  703 FORMAT(/27H STOPPED IN FRONT. MSTIF (=,I4,21H) EXCEEDED IN ELEMENT
     1,I4, 22H. NEEDS TO BE AT LEAST,I5)
  705 FORMAT(/26H STOPPED IN FRONT, ELEMENT,I4,19H IMPERMISSIBLE TYPE)
      END
C
C
C
      SUBROUTINE OUTPUT(MTOTV,MPOIN,MELEM,MEVAB,MNODE,MSETS,MCOMP,
     1MPOIN,NELEM,NDIME,NGAUS,NSTRE,MSTYP,NDOFM,NPROP,
     2LNODS,LNODN,LSIDN,NPNDF,LMTYP,LOREQ,LPROS,COORD,ENCOD,SHAPE,
     3PROPS,ENPRO,DERIV,CARTD,BMATX,ASLOD,ASDIS,ENDIS,ASDS1,
     4SCONV,RMAX,RMSR,INC,IT,ISUPN,PSCAL,PSTSL,STRSL,RFACT,INDPT)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION LNODS(MELEM,MNODE),LNODN(MELEM),LSIDN(MELEM),LTYPS(9)
     1,NPNDF(MPOIN),LMTYP(MELEM),LOREQ(MELEM),LPROS(MELEM,MNODE)
     2,COORD(MPOIN,3),ENCOD(3,MNODE),GPCOD(3),GPLOC(3,8),GPWTS(8)
     3,SHAPE(MNODE),DERIV(3,MNODE),CARTD(3,MNODE),BMATX(6,MEVAB)
     4,DMATX(6,6),PROPS(MSETS,MCOMP),ENPRO(MCOMP,MNODE)
     5,ASLOD(MTOTV),ASDIS(MTOTV),ENDIS(MEVAB),ASDS1(MTOTV)
     6,FORCE(6),DISPL(6),AVCOD(3),STRES(6),STRIN(6),PSTRS(12),PSTRN(12)
     7,AVSTS(6),AVSTN(6),AVPSS(12),AVPSN(12),PSTSL(MELEM,4)
     8,STRSL(MELEM,6),INDPT(16)
C
      DATA IBRAC,JBRAC,SMALL/1H(,1H),1.0D-12/
      DATA LTYPS/21,31,33,44,63,64,84,86,206/
C
C
C...FIND NDSTR; IF DOF.GT.NDSTR IT IS A PORE-PRESSURE D.O.F.
```

```
      NDSTR = 2
      IF (NSTYP.GE.2) NDSTR = 3
      WRITE(6,600) INC,IT
      IF(ISUPN.LT.0) GO TO 17
      NODSUP = INDPT(1)
C
C...OUTPUT NODAL FORCES AND DISPLACEMENTS
C...(IF MORE THAN 3 D.OF F. PER NODE ALTER FORMATS 610 - 615)
      WRITE(6,610)
      WRITE(6,611)
      WRITE(6,612)(IBRAC,I,JBRAC,I=1,NDOFN)
      WRITE(6,613)(IBRAC,I,JBRAC,I=1,NDOFN)
      NIK = 0
      DO 15 NP=1,NPOIN
      NDOFN = NPNDF(NP)
C
      IF (NODSUP.EQ.0) GO TO 10
      DO 7 I=1,16
      IF (INDPT(I).EQ.NP) GO TO 10
    7 CONTINUE
      GO TO 14
   10 CONTINUE
C
      DO 12 N=1,NDOFN
      NIKN = NIK + N
      FORCE(N) = ASLOD(NIKN)
      DISPL(N) = ASDIS(NIKN)
      IF (N.GT.NDSTR) DISPL(N) = DISPL(N)*PSCAL
   12 CONTINUE
      WRITE(6,615) NP,(FORCE(N),N=1,NDOFN)
      WRITE(6,616) (DISPL(N),N=1,NDOFN)
   14 NIK = NIK + NDOFN
   15 CONTINUE
      WRITE(6,617) RMAX,RMSR
C
C
C...SET NSTPD FOR 3D PRINC DIRECTION OUTPUT
   17 NSTPD = NSTRE
      IF(NSTYP.EQ.3) NSTPD=12
C...ELEMENT OUTPUT ACCORDING TO INSTRUCTIONS IN LOREQ()
      DO 140 L=1,NELEM
      LORQL = LOREQ(L)
      IF (LORQL.EQ.0.AND.LMTYP(L).LE.2) GO TO 140
      NNODE = LNODN(L)
      NSIDE = LSIDN(L)
      MATYP = LMTYP(L)
      NODSID = NNODE*10 + NSIDE
      DO 20 LTYPE=1,9
      IF(NODSID.EQ.LTYPS(LTYPE)) GO TO 21
   20 CONTINUE
C...IF DO 20 COMPLETED ELEMENT TYPE IS IMPERMISSIBLE
      GO TO 300
C...SET UP HEADING
   21 IF (LORQL.NE.0) WRITE(6,620) L,NNODE,NSIDE,MATYP
C...SEPERATE LOREQ() INTO 4 1 FIG. INTEGERS
      LORQ1 = LORQL/1000
      LORQ2 = LORQL/100 - LORQ1*10
      LORQ3 = LORQL/10 - (LORQL/100)*10
      LORQ4 = LORQL - (LORQL/10)*10
      IF (LORQL.EQ.0) GO TO 23
```

```
C...MORE HEADINGS
      WRITE(6,621)
      WRITE(6,622)
      WRITE(6,623) (IBRAC,I,JBRAC,I=1,NDIME)
      WRITE(6,624) (IBRAC,I,JBRAC,I=1,NSTRE)
      WRITE(6,625)
C
C...SET UP EL. NODE CO-ORD AND DISP. ARRAYS
   23 CONTINUE
      NEVAB = 0
      DO 30 N=1,NNODE
      NP = LNODS(L,N)
      DO 25 M=1,3
   25 ENCOD(M,N) = COORD(NP,M)
      K = LPROS(L,N)
      DO 26 M=1,NPROP
   26 ENPRO(M,N) = PROPS(K,M)
      NIK = NPOSN(NP,0,NPNDF)
      NDOFN = NPNDF(NP)
      DO 27 M=1,NDSTR
      NEVAB = NEVAB+1
      NIKM = NIK + M
   27 ENDIS(NEVAB) = ASDIS(NIKM) - ASDS1(NIKM)
   30 CONTINUE
C
C...ZERO ARRAYS TO CONTAIN GAUSS POINT AVERAGES
      DO 35 M=1,NDIME
   35 AVCOD(M) = 0.D0
      DO 37 I=1,NSTRE
      AVSTS(I) = 0.D0
   37 AVSTN(I) = 0.D0
      DO 38 I=1,NSTPD
      AVPSS(I) = 0.D0
   38 AVPSN(I) = 0.D0
      AVEYN = 0.0D0
C
      CALL GAUSS(NGAUS,LTYPE,NGTOT,GPWTS,GPLOC)
C...ENTER GAUSS LOOP
      DO 120 IJ=1,NGTOT
      S = GPLOC(1,IJ)
      T = GPLOC(2,IJ)
      V = GPLOC(3,IJ)
      CALL SFR(S,T,V,LTYPE,MNODE,SHAPE,DERIV)
      DO 45 M=1,NDIME
      SUM = 0.D0
      DO 43 N=1,MNODE
   43 SUM = SUM + ENCOD(M,N)*SHAPE(N)
      GPCOD(M) = SUM
   45 AVCOD(M) = AVCOD(M) + SUM*GPWTS(IJ)
C
C...OBTAIN B AND THEN D MATRICES
      CALL JACOB(MNODE,MNODE,LTYPE,NDIME,L,ENCOD,SHAPE,DERIV,CARTD,
     1DJACB)
      IF (LTYPE.GT.2) GO TO 47
      CALL BARB(MEVAB,MNODE,MNODE,ENCOD,SHAPE,DERIV,CARTD,DJACB,
     1SCONV,BMATX)
      GO TO 49
   47 R = GPCOD(1)
      IF(R.LT.SMALL) R=SMALL
      CALL BMAT(MPOIN,MELEM,MNODE,MEVAB,MNODE,MSTYP,L,LNODS,NPNDF
```

```
     1,R,SHAPE,CARTD,SCONV,BMATX)
   40 YM = 0.D0
      PR = 0.D0
      DO 50 N=1,MNODE
      YMN = ENPRO(1,N)
      IF (LMTYP(L).EQ.3) CALL YMNYP(MELEM,MNODE,MSETS,MCOMP,L,N,
     1PSTSL,LPROS,PROPS,RFACT,YMN)
      IF (LMTYP(L).EQ.4) CALL YMLIN(MELEM,MNODE,MSETS,MCOMP,L,N,
     1PSTSL,LPROS,PROPS,YMN)
      YM = YM + YMN*SHAPE(N)
      PR = PR + ENPRO(2,N)*SHAPE(N)
   50 CONTINUE
      CALL MOD1(MSTYP,NSTRE,YM,PR,BMATX)
C...CALC STRESSES AND STRAINS
      DO 65 I=1,NSTRE
      STRSI = 0.D0
      STRNI = 0.D0
      DO 60 N=1,NEVAB
      SUM = 0.D0
      DO 58 J=1,NSTRE
   58 SUM = SUM + BMATX(J,I)*BMATX(J,N)
      STRSI = STRSI + SUM*ENDIS(N)
   60 STRNI = STRNI + BMATX(I,N)*ENDIS(N)
      STRES(I) = STRSI
      STRIN(I) = STRNI
      AVSTS(I) = AVSTS(I) + STRSI*GPWTS(IJ)
   65 AVSTN(I) = AVSTN(I) + STRNI*GPWTS(IJ)
      AVEYM = AVEYM + YM*GPWTS(IJ)
C
C...OUTPUT G.P. NO., CO-ORDS - ALL CASES UNLESS LORQ1=0
      IF(LORQ1.EQ.0) GO TO 80
      WRITE(6,630) IJ,(GPCOD(M),M=1,NDIME)
      IF(LORQ2.EQ.0) GO TO 80
C...OUTPUT STRESSES FOLLOWED BY STRAINS IF REQD.
      WRITE(6,631)(STRES(I),I=1,NSTRE)
      WRITE(6,625)
      IF(LORQ2.NE.2) GO TO 80
      WRITE(6,632)(STRIN(I),I=1,NSTRE)
      WRITE(6,625)
C
C...COMPUTE AND OUTPUT PRINC. STRESSES (LORQ3=1 OR 2)
   80 IF(LORQ3.EQ.0) GO TO 100
      CALL PRINC(1,NSTRE,STRES,PSTRS)
      DO 85 I=1,NSTPD
   85 AVPSS(I) = AVPSS(I) + PSTRS(I)*GPWTS(IJ)
      IF(LORQ1.EQ.0) GO TO 90
      WRITE (6,660) YM
      WRITE(6,633) (PSTRS(I),I=1,NSTPD)
      WRITE(6,625)
C
C...COMPUTE AND OUTPUT PRINC. STRAINS (LORQ3=2)
   90 IF(LORQ3.NE.2) GO TO 100
      CALL PRINC(2,NSTRE,STRIN,PSTRN)
      DO 95 I=1,NSTPD
   95 AVPSN(I) = AVPSN(I) + PSTRN(I)*GPWTS(IJ)
      IF(LORQ1.EQ.0) GO TO 100
      WRITE(6,634) (PSTRN(I),I=1,NSTPD)
      WRITE(6,625)
C
C...COMPUTE AND OUTPUT INVARIANTS - EXCEPT THAT WE LEAVE THIS TO YOU!
```

```
  100 IF(LORQ4.NE.0) WRITE(6,635)
C
  120 CONTINUE
C...END OF GAUSS LOOP
C
C
C...OUTPUT G.P. AVERAGES
      NGD = (1 + NSIDE)/2
      AA = DFLOAT(NGAUS**NGD)
      A = 1./AA
      IF(NSIDE.EQ.3) A=2.
      DO 123 M=1,NDIME
  123 AVCOD(M) = AVCOD(M)*A
      DO 125 I=1,NSTRE
      AVSTS(I) = AVSTS(I)*A
      AVSTN(I) = AVSTN(I)*A
      AVPSS(I) = AVPSS(I)*A
      AVPSN(I) = AVPSN(I)*A
  125 CONTINUE
      AVEYM = AVEYM*A
C...ADD STRESS INCREMENTS TO STRSL.
      DO 126 I=1,NSTRE
      AVSTS(I) = AVSTS(I) + STRSL(L,I)
  126 STRSL(L,I) = AVSTS(I)
      IF (LORQL.EQ.0) GO TO 128
C
      WRITE(6,640) (AVCOD(M),M=1,NDIME)
C...G.P.AV STRESSES, AND STRAINS IF REQD.
      IF (LORQ2.EQ.0) GO TO 128
      WRITE (6,631) (AVSTS(I),I=1,NSTRE)
C...WRITE AVE. PRINC. EFFECTIVE STRESSES TO PSTSL.
  128 CONTINUE
      CALL PRINC(1,NSTRE,AVSTS,AVPSS)
      MINOR = 2
      IF (NSTRE.EQ.6) MINOR = 3
      PSTSL(L,1) = PSTSL(L,3)
      PSTSL(L,2) = PSTSL(L,4)
      PSTSL(L,3) = AVPSS(1)
      PSTSL(L,4) = AVPSS(MINOR)
      IF (LORQL.EQ.0) GO TO 135
      WRITE(6,625)
      IF(LORQ2.NE.2) GO TO 130
      WRITE(6,632) (AVSTN(I),I=1,NSTRE)
      WRITE(6,625)
C...G.P. AV PRINC. STRESSES, AND STRAINS IF REQD.
  130 IF(LORQ3.EQ.0) GO TO 135
C
      WRITE (6,660) AVEYM
      WRITE(6,633) (AVPSS(I),I=1,NSTRE)
      WRITE(6,625)
      IF(LORQ3.NE.2) GO TO 135
      WRITE(6,633) (AVPSN(I),I=1,NSTRE)
      WRITE(6,625)
C...G.P.AV INVARIANTS - THIS SECTION TO BE ADDED
  135 CONTINUE
C
  140 CONTINUE
C...END OF ELEMENT LOOPS
C
C
```

```
  150 WRITE (6,650) INC,IT
      RETURN
C
C...ERROR MESSAGE
  300 WRITE(6,700) L
      STOP
C
C...OUTPUT FORMATS
  600 FORMAT(//1X,40(1H*),21H OUTPUT FOR INCREMENT,I4,10H ITERATION,
     1 I4,1H ,40(1H*)//)
  610 FORMAT(/2X,4HNODE,3X,22HLOAD (INCL. REACTIONS))
  611 FORMAT(1H+,53X,13HDISPLACEMENTS)
  612 FORMAT(3X,4HNO. ,3(5X,A1,I1,A1,5X))
  613 FORMAT(1H+,48X,3(5X,A1,I1,A1,5X))
  615 FORMAT(I5,2X,3D13.5)
  616 FORMAT(1H+,48X,3D13.5)
  617 FORMAT(/24H FORCE RESIDUALS, RMAX=,D13.5,/
     1 1X,15(1H-),3H  RMSR =,D13.5//)
  620 FORMAT( 12X,11HELEMENT NO.,I4,2H (,I2,7H NODES,,I2,24H SIDES)  MAT
     1ERIAL TYPE =,I4,/12X,56(1H-))
  621 FORMAT(6H GAUSS, 7X,12HG.P. CO-ORDS)
  622 FORMAT(1H+,50X,24HSTRESS/STRAIN COMPONENTS)
  623 FORMAT(6H POINT,1X,3(4X,A1,I1,A1,3X))
  624 FORMAT(1H+,42X,6(5X,A1,I1,A1,5X))
  625 FORMAT(1H )
  630 FORMAT(1H+,I3,2X,3F10.3)
  631 FORMAT(1H+,38X,4HSTRS,6D13.5)
  632 FORMAT(1H+,38X,4HSTRN,6D13.5)
  633 FORMAT(1H+,38X,4HPSTS,3D13.5,/34X,9HDIR. COS.,3(1X,3F8.3,1X))
  634 FORMAT(1H+,38X,4HPSTN,3D13.5,/34X,9HDIR. COS.,3(1X,3F8.3,1X))
  635 FORMAT(10X,40HINVARIANTS REQD. - YOU HAVE TO CODE THIS)
  640 FORMAT(1H+,5H G.AV,3F10.3)
  650 FORMAT(/1X,4(1H*),24H END OF OUTPUT FOR INC =,I4,5H IT =,I4,1H ,
     1 41(1H*))
  660 FORMAT(1H+,83X,2HYH,D13.5)
  700 FORMAT(/27H STOPPED IN OUTPUT, ELEMENT,I4,19H IMPERMISSIBLE TYPE)
      END
C
C
      SUBROUTINE SMOOTH(MTOTV,MPOIN,MFIXV,MSTYP,MELEM,MNODE,
     1MELEM,MPOIN,NSDIS,LSIDN,ASDIS,BSDIS,COORD,NPNDF,SPDIS,
     2SUMS,SHAPE,ENCOD,LNODS,LNODN,LMTYP,NFVAR,NBVAR,NBDIS,MPFIX)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ASDIS(MTOTV),COORD(MPOIN,3),NFVAR(MFIXV),
     1NPNDF(MPOIN),SPDIS(MFIXV),LSIDN(MELEM),BSDIS(MPFIX),
     2LNODS(MELEM,MNODE),LNODN(MELEM),LMTYP(MELEM),LTYPS(9),
     3NBVAR(MPFIX),SUMS(MPOIN,2),SHAPE(MNODE),ENCOD(3,MNODE)
C
      DATA LTYPS/21,31,33,44,63,64,84,86,206/
C
C...SMOOTHS PWPS BY ELTS, PUTTING SURFACES THRO' G.P. AVES.
C
      NDSTR = 2
      IF (MSTYP.GE.2) NDSTR=3
      DO 5 I=1,NSDIS
      BSDIS(I) = SPDIS(I)
    5 NBVAR(I) = NFVAR(I)
      DO 10 I=1,NPOIN
      SUMS(I,1) = 0.0D0
```

```
   10 SUMS(I,2) = 0.0D0
C...LOOP OVER ALL ELEMENTS
      DO 80 L=1,NELEM
C...IF ELT IS STRUCTURAL, SKIP IT.
      IF (LMTYP(L).EQ.1) GO TO 80
      NNODE = LNODN(L)
      NSIDE = LSIDN(L)
      NODSID = NNODE*10 + NSIDE
      DO 15 LTYPE=1,9
      IF (NODSID.EQ.LTYPS(LTYPE)) GO TO 20
   15 CONTINUE
      GO TO 910
C...PUT COORDS & P.W.P.S INTO ENCOD &  SHAPE .
   20 DO 25 N=1,NNODE
      NN = LNODS(L,N)
      NDOFN = NPNDF(NN)
      NIKNO = NPOSN(NN,NDOFN,NPNDF)
      SHAPE(N) = ASDIS(NIKNO)
      DO 23 J=1,3
   23 ENCOD(J,N) = COORD(NN,J)
   25 CONTINUE
      GO TO (910,910,27,27,55,45,50,910,910), LTYPE
C...FIND AREA FOR LINEAR TRIANGLE OR QUADRILATERAL
   27 SA = DSQRT((ENCOD(1,2)-ENCOD(1,1))**2+(ENCOD(2,2)-ENCOD(2,1))**2)
      SB = DSQRT((ENCOD(1,3)-ENCOD(1,2))**2+(ENCOD(2,3)-ENCOD(2,2))**2)
      SC = DSQRT((ENCOD(1,1)-ENCOD(1,3))**2+(ENCOD(2,1)-ENCOD(2,3))**2)
      SEMIP = (SA + SB + SC)/2.
      AREA = DSQRT(SEMIP*(SEMIP-SA)*(SEMIP-SB)*(SEMIP-SC))
      IF (LTYPE.EQ.3) GO TO 30
C...FOR QUADRILATERAL, ADD ON AREA OF SECOND TRIANGLE
      SA = DSQRT((ENCOD(1,4)-ENCOD(1,3))**2+(ENCOD(2,4)-ENCOD(2,3))**2)
      SB = DSQRT((ENCOD(1,1)-ENCOD(1,4))**2+(ENCOD(2,1)-ENCOD(2,4))**2)
      SEMIP = (SA + SB + SC)/2.
      AREA = AREA + DSQRT(SEMIP*(SEMIP-SA)*(SEMIP-SB)*(SEMIP-SC))
   30 CONTINUE
C...NOW AVERAGE P.W.P.S AND STORE IN SHAPE
      AVEPWP = 0.0D0
      DO 35 N=1,NNODE
      AVEPWP = AVEPWP + SHAPE(N)
   35 CONTINUE
      AVEPWP = AVEPWP/DFLOAT(NNODE)
      DO 40 N=1,NNODE
      SHAPE(N) = AVEPWP
   40 CONTINUE
      GO TO 60
C...SIX-NODED RECTANGLE
   45 CONTINUE
      AREA = (ENCOD(1,6)-ENCOD(1,1))**2 + (ENCOD(2,6)-ENCOD(2,1))**2
      AREA=AREA*((ENCOD(1,3)-ENCOD(1,1))**2+(ENCOD(2,3)-ENCOD(2,1))**2)
      AREA = DSQRT(AREA)
      PWP1 = (SHAPE(1)+SHAPE(6)+SHAPE(2)+SHAPE(5))/3. -
     1 (SHAPE(3)+SHAPE(4))/6.
      PWP3 = (SHAPE(3)+SHAPE(4)+SHAPE(2)+SHAPE(5))/3. -
     1 (SHAPE(1)+SHAPE(6))/6.
      SHAPE(1) = PWP1
      SHAPE(2) = (PWP1 + PWP3)/2.
      SHAPE(3) = PWP3
      SHAPE(4) = PWP3
      SHAPE(5) = (PWP1 + PWP3)/2.
      SHAPE(6) = PWP1
```

```
      GO TO 60
C...EIGHT-NODED QUADRILATERAL
   50 CONTINUE
      AREA = (ENCOD(1,3)-ENCOD(1,1))**2 + (ENCOD(2,3)-ENCOD(2,1))**2
      AREA=AREA+(ENCOD(1,5)-ENCOD(1,3))**2+(ENCOD(2,5)-ENCOD(2,3))**2
      AREA = DSQRT(AREA)
      PMID = DMAX1((SHAPE(2)+SHAPE(6))/2.,(SHAPE(4)+SHAPE(8))/2.)
      PWP1 = (4.*(SHAPE(1)+SHAPE(2)+SHAPE(8))-2.*(SHAPE(3)+SHAPE(7)
     1+SHAPE(4)+SHAPE(6))+SHAPE(5)+4.*PMID)/9.
      PWP3 = (4.*(SHAPE(3)+SHAPE(2)+SHAPE(4))-2.*(SHAPE(1)+SHAPE(5)
     1+SHAPE(6)+SHAPE(8))+SHAPE(7)+4.*PMID)/9.
      PWP5 = (4.*(SHAPE(5)+SHAPE(4)+SHAPE(6))-2.*(SHAPE(3)+SHAPE(7)
     1+SHAPE(2)+SHAPE(8))+SHAPE(1)+4.*PMID)/9.
      PWP7 = (4.*(SHAPE(7)+SHAPE(6)+SHAPE(8))-2.*(SHAPE(1)+SHAPE(5)
     1+SHAPE(2)+SHAPE(4))+SHAPE(3)+4.*PMID)/9.
      SHAPE(1) = PWP1
      SHAPE(2) = (PWP1 + PWP3)/2.
      SHAPE(3) = PWP3
      SHAPE(4) = (PWP5 + PWP3)/2.
      SHAPE(5) = PWP5
      SHAPE(6) = (PWP7 + PWP5)/2.
      SHAPE(7) = PWP7
      SHAPE(8) = (PWP1 + PWP7)/2.
      GO TO 60
C...SIX-NODED EQUILATERAL TRIANGLE
   55 CONTINUE
      SA = DSQRT((ENCOD(1,3)-ENCOD(1,1))**2+(ENCOD(2,3)-ENCOD(2,1))**2)
      SB = DSQRT((ENCOD(1,5)-ENCOD(1,3))**2+(ENCOD(2,5)-ENCOD(2,3))**2)
      SC = DSQRT((ENCOD(1,1)-ENCOD(1,5))**2+(ENCOD(2,1)-ENCOD(2,5))**2)
      SEMIP = (SA+SB+SC)/2.
      AREA = DSQRT(SEMIP*(SEMIP-SA)*(SEMIP-SB)*(SEMIP-SC))
      PWP1 = SHAPE(2) - SHAPE(4) + SHAPE(6)
      PWP3 = SHAPE(2) + SHAPE(4) - SHAPE(6)
      PWP5 =-SHAPE(2) + SHAPE(4) + SHAPE(6)
      SHAPE(1) = PWP1
      SHAPE(2) = (PWP1 + PWP3)/2.
      SHAPE(3) = PWP3
      SHAPE(4) = (PWP5 + PWP3)/2.
      SHAPE(5) = PWP5
      SHAPE(6) = (PWP5 + PWP1)/2.
C...NOW ADD SMOOTHED PWPS*AREA AND AREAS TO SUMS
   60 CONTINUE
      DO 65 N=1,NNODE
      NN = LNODS(L,N)
      SUMS(NN,1) = SUMS(NN,1) + SHAPE(N)*AREA
      SUMS(NN,2) = SUMS(NN,2) + AREA
   65 CONTINUE
   80 CONTINUE
C
C...DIVIDE SUMS*AREAS BY AREAS FOR FINAL SMOOTHED PWPS
      NBDIS = NSDIS
      DO 85 N=1,NPOIN
      NDOFN = NPNDF(N)
      IF (NDOFN.LE.NDSTR) GO TO 85
      NIKNO = NPOSN(N,NDOFN,NPNDF)
      DO 82 I=1,NSDIS
   82 IF (NIKNO.EQ.NFVAR(I)) GO TO 85
      NBDIS = NBDIS + 1
      IF (NBDIS.GT.NPFIX) GO TO 900
      DSDIS(NBDIS) = SUMS(N,1)/SUMS(N,2)
```

```
      KBVAR(NBDIS) = NIKNO
   85 CONTINUE
C
      RETURN
  900 WRITE (6,599)
  599 FORMAT(1H0,34HSTOPPED IN SMOOTH. MPFIX EXCEEDED./
     11X,53HMPFIX MUST EXCEED MFIXV + NO. OF PORE-PRESSURE NODES.)
      STOP
  910 WRITE (6,598) L
  598 FORMAT(1H0,26HSTOPPED IN SMOOTH. ELEMENT,I5,15H IS WRONG TYPE.)
      STOP
      END
C
C
C
```

(F0000) PRINT COMPLETED 1071 LINES

Program Listing

Program SUBCOM(F)

This contains the satellite subroutines which are common to FINEPAK and FINETIM. The only additions are subroutines to evaluate the Young's modulus in a non-linear element; the other subroutines have been omitted.

```
      SUBROUTINE YHYP(MELEM,MNODE,MSETS,MCOMP,L,N,PSTSL,LPROS,
     1PROPS,RFACT,YH)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LPROS(MELEM,MNODE),PROPS(MSETS,MCOMP),PSTSL(MELEM,4)
C
C...FINDS YOUNG'S MOD FOR NODE N OF HYPERBOLIC ELT L
C
      K = LPROS(L,N)
      YH = PROPS(K,1)
      CC = PROPS(K,3)
      PHI = PROPS(K,4)/57.2958D0
      S1 = PSTSL(L,3)
      S3 = PSTSL(L,4)
      DEVOLD = PSTSL(L,1) - PSTSL(L,2)
      DEVNEW = S1 - S3
      IF (DEVNEW.LE.DEVOLD) RETURN
      SINP = DSIN(PHI)
      COSP = DCOS(PHI)
      FACT = 1. - RFACT*(S1-S3)*(1.-SINP)/(2.*S3*SINP+2.*CC*COSP)
      IF (FACT.LE.0.1D0) FACT = 0.1D0
      YH = YH*FACT*FACT
      RETURN
      END
C
C
      SUBROUTINE YHLIN(MELEM,MNODE,MSETS,MCOMP,L,N,PSTSL,LPROS,
     1PROPS,YH)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LPROS(MELEM,MNODE),PROPS(MSETS,MCOMP),PSTSL(MELEM,4)
C
C...EVALUATES YOUNG'S MOD FOR NODE N OF BILINEAR ELT L
C
      K = LPROS(L,N)
      YH = PROPS(K,1)
      COEFF = PROPS(K,3)
```

```
      DEVOLD = PSTSL(L,1) - PSTSL(L,2)
      DEVNEW = PSTSL(L,3) - PSTSL(L,4)
      IF (DEVNEW.GE.DEVOLD) RETURN
      YH = YH*COEFF
      RETURN
      END
C
C
```